

# Ordonnancement des processus

---

SYSTÈMES D'EXPLOITATION  
AVANCÉES

# La gestion des processus

---

Un processus est un programme en cours d'exécution.

# Avantages des processus

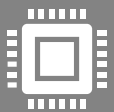
---



Le fractionnement des applications qui peut en simplifier le développement,



l'avantage pour l'utilisateur de savoir faire tourner plusieurs applications simultanément



et surtout l'optimisation de l'utilisation du (ou des) processeurs(s)

# A quoi sert un processus

---

faire plusieurs activités "en même temps".

- Exemples
  - Faire travailler plusieurs utilisateurs sur la même machine. Compiler tout en lisant son mail
- Problème: Un processeur ne peut exécuter qu'une seule instruction à la fois.
- BUT: Partager un (ou plusieurs) processeur entre différents processus.

**Attention!!! Ne pas confondre processus et processeur**

# Bloc de contrôle de processus

---

Chaque processus est représenté dans le SE par un PCB

PCB	
Pointeur	État du processus
Numéro du processus	
Compteur d'instructions	
Registres	
Limite de la mémoire	
Liste des fichiers ouverts	
...	

# Le PCB

---

PCB: contient plusieurs informations concernant un processus spécifique, comme par exemple:

- L'état du processus.
- Compteur d'instructions: indique l'adresse de l'instruction suivante devant être exécutée par ce processus.
- Informations sur le scheduling de la CPU: information concernant la priorité du processus.
- Informations sur la gestion de la mémoire: valeurs des registres base et limite, des tables de pages ou des tables de segments.
- Informations sur l'état des E/S: liste des périphériques E/S allouées à ce processus, une liste des fichiers ouverts, etc.

## Critères d'Ordonnancement

Utilisation du CPU

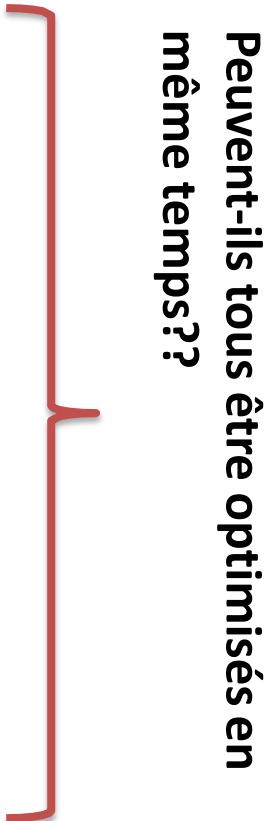
Débit (Throughput)

Temps de rotation (Turnaround  
time)

Temps d'attente

Temps de réponse

# Critères d'optimisation: max/min

- Les critères d'ordonnancement peuvent être ramenés à des problèmes d'optimisation. Chaque critère devra être maximisé ou minimisé.
  - Utilisation du CPU
    - Maximiser
  - Débit (Throughput)
    - Maximiser
  - Temps de rotation (Turnaround time)
    - Minimiser
  - Temps d'attente
    - Minimiser
  - Temps de réponse
    - Minimiser
- 
- Peuvent-ils tous être optimisés en même temps??



## Avec/Sans réquisition

- **Ordonnancement sans réquisition (non préemptif):**
  - Une fois que le CPU a été allouée à un processus, ce dernier le garde jusqu'à ce qu'il le libère, soit parce qu'il a terminé, soit parce qu'il commute à l'état en attente.
  - Algorithmes simples et faciles à mettre en œuvre mais pas adaptés pour le temps partagé et seulement pour les système de traitement par lots.
- **Ordonnancement avec réquisition (préemptif):**
  - Un processus peut être suspendu à n'importe quel instant, sans avoir été prévenu, pour laisser la place à un autre processus.
  - Implémentation plus couteuse à cause de la commutation de processus, utilisée dans les systèmes à temps partagé et systèmes temps-réels.
  - Nécessité d'un mécanisme de synchronisation.



# Ordonnancement sans réquisition

# FIFO (First In First Out)

- Allocation dans l'ordre d'arrivée (premier arrivé = premier servi)

<u>Processus</u>	<u>durée</u>
$P_1$	24
$P_2$	3
$P_3$	3

- L'ordre d'arrivée est:  $P_1, P_2, P_3$



- Le temps d'attente pour  $P_1 = 0$ ;  $P_2 = 24$ ;  $P_3 = 27$
- Temps moyen d'attente:  $(0 + 24 + 27)/3 = 17$

# SJF

---

- ❖ Dans cette méthode, le processeur est affecté au processus ayant le plus petit temps d'exécution
- ❖ Cela nécessite une connaissance future du temps d'exécution.
- ❖ Dans nos exemples, il est donné sous forme de tableau mais en réalité ces temps ne sont pas connus par le système d'exploitation. Il fait donc des prévisions.
- ❖ Une approche pour cette prédiction consiste à utiliser les temps de d'exécution de processeur précédents pour les processus dans la file d'attente des prêts, puis l'algorithme sélectionne le temps de processeur suivant prévu le plus court.

# Les Algorithmes d'ordonnancement (SJF + PAPS)

---

## Exemple 1.

- Considérons cinq travaux A, B, C, D et E, dont les temps d'exécution et leurs arrivages respectifs sont donnés dans la table Faire un schéma qui illustre son exécution et calculer le temps de séjour de chaque processus, le temps moyen de séjour, le temps d'attente et le temps moyen d'attente en utilisant :

1. Premier arrivé premier servi (**PAPS**)
2. Le plus court d'abord (**SJF**)

# Les Algorithmes d'ordonnancement (SJF + PAPS)

---

Processus	Temps d'exécution	Temps d'arrivée
A	3	0
B	6	1
C	4	4
D	2	6
E	1	7

A	A	A	B	B	B	B	B	B	C	C	C	C	D	D	E
1				5					10					15	

# Les Algorithmes d'ordonnancement (SJF + PAPS)

---

Processus	Temps de sejour
A	$3-0 = 3$
B	$9-1 = 8$
C	$13-4 = 9$
D	$15-6 = 9$
E	$16-7 = 9$

Le temps moyen de séjour est :  $\frac{(3+8+9+9+9)}{5} = \frac{38}{5} = 7,6$

# Les Algorithmes d'ordonnancement (SJF + PAPS)

---

Le temps d'attente est calculé soustrayant le temps d'exécution du temps de séjour :

Processus	Temps d'attente
A	$3-3 = 0$
B	$8-6 = 2$
C	$9-4 = 5$
D	$9-2 = 7$
E	$9-1 = 8$

Le temps moyen d'attente est :  $\frac{(0+2+5+7+8)}{5} = \frac{23}{5} = 4,4$

Il y a cinq tâches exécutées dans 16 unités de temps, alors  $16/5 = 3,2$  unités de temps par processus.



# Les Algorithmes d'ordonnancement (SJF + PAPS)

## 2. SJF

A	A	A	B	B	B	B	B	B	E	D	D	C	C	C	C
1				5					10					15	

Pour la stratégie SJF nous aurons la séquence d'exécution A,B,E,D,C, et le temps de séjour est :

Processus	Temps de séjour
A	$3-0 = 3$
B	$9-1 = 8$
E	$10-7 = 3$
D	$12-6 = 6$
C	$16-4 = 12$

$$\frac{(3+8+3+6+12)}{5} = \frac{32}{5} = 6,4$$

# Les Algorithmes d'ordonnancement (SJF + PAPS)

---

Processus	Temps d'attente
A	$3-3 = 0$
B	$8-6 = 2$
E	$3-1 = 2$
D	$6-2 = 4$
C	$12-4 = 8$

**Le temps moyen d'attente est :**  $\frac{(0+2+2+4+8)}{5} = \frac{16}{5} = 3,2$

**Il y a cinq tâches exécutées dans 16 unités de temps, alors  $16/5 = 3,2$  unités de temps par processus.**

# Algorithme du Tourniquet (Round Robin)

- Algorithme conçu spécialement pour le temps partagé
- Les processus accèdent au processeur, chacun à leur tour, pour un temps maximal déterminé à l'avance (le quantum noté  $q$  = en général 10-100 millisecondes),
- Lorsqu'il a épuisé ce temps, ou qu'il se bloque : le processus suivant est élu et le remplace.
- Le processus suspendu est mis en queue du tourniquet (file FIFO circulaire).

# Exemple

---

Soient deux processus A et B prêts tels que A est arrivé en premier suivi de B, 2 unités de temps après. Les temps de UCT nécessaires pour l'exécution des processus A et B sont respectivement 15 et 4 unités de temps. Le temps de commutation est supposé nul. Calculer le temps de séjour de chaque processus A et B, le temps moyen de séjour, le temps d'attente, le temps moyen d'attente, et le nombre de changements de contexte pour :

- SRT
- Round robin (quantum = 10 unités de temps)
- Round robin (quantum = 3 unités de temps)

# Solution SRT :

---

A	A	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A
1				5					10					15				

Processus	Temps de séjour
A	$19 - 0 = 19$
B	$6 - 2 = 4$

**Temps moyen de séjour** =  $\frac{19+4}{2} = 11,5$

Processus	Temps d'attente
A	$19 - 15 = 4$
B	$4 - 4 = 0$

**Le temps moyen d'attente** =  $\frac{4+0}{2} = 2$

Il y a 3 changements de contexte.

# Round robin (quantum = 10)

A	A	A	A	A	A	A	A	A	A	B	B	B	B	A	A	A	A	A
1				5					10					15				

Processus	Temps de séjour
A	19-0 = 19
B	14-2 = 12

Temps moyen de séjour =  $\frac{19+12}{2} = 15,5$

Processus	Temps d'attente
A	19-15 = 4
B	12-4 = 8

Le temps moyen d'attente =  $\frac{4+8}{2} = 6$

<sup>2</sup>Le quantum était de 1 seconde dans les premières versions d'Unix.

Il y a 3 changements de contexte.

# Round robin (quantum = 3)

Round robin (quantum = 3) :

A	A	A	B	B	B	A	A	A	B	A	A	A	A	A	A	A	A	A
1				5					10					15				

Processus	Temps de séjour
A	$19 - 0 = 19$
B	$10 - 2 = 8$

**Temps moyen de séjour** =  $\frac{19+8}{2} = 13,5$

Processus	Temps d'attente
A	$19 - 15 = 4$
B	$8 - 4 = 4$

**Le temps moyen d'attente** =  $\frac{4+4}{2} = 4$

Il y a 5 changements de contexte.

Dans le trois solutions (SRT, RR  $Q_t=10$  et RR  $Q_t=3$ ), il y a 2 tâches exécutées dans 19 unités de temps, alors  $19/2 = 9,5$  **unités de temps par processus**.

# L'ordonnancement à priorité

---

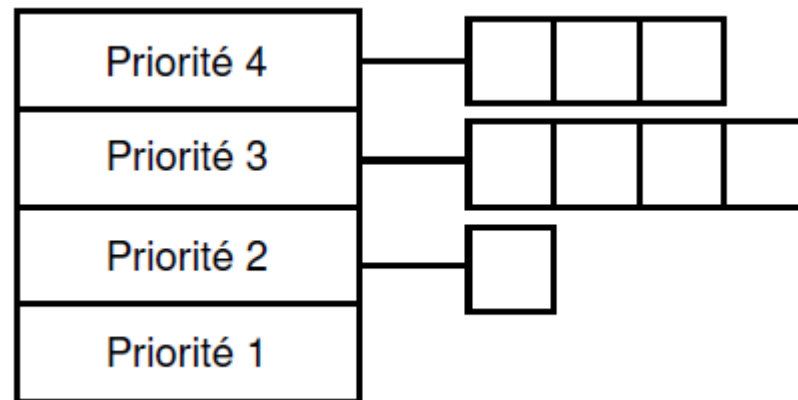
L'**ordonnanceur à priorité attribue à chaque** processus une priorité. Le choix du processus à élire dépend des priorités des processus prêts.

Les processus de même priorité sont regroupés dans une le du type FIFO. Il y a autant de les qu'il y a de niveaux de priorité. L'ordonnanceur choisit le processus le plus prioritaire qui se trouve en tête de le. En général, les processus de même priorité sont ordonnancés selon l'algorithme du tourniquet.



# L'ordonnancement à priorité

---



# Exemple

Pour les processus du tableau suivant, dessinez un schéma illustrant leur exécution, en utilisant l'ordonnancement avec priorités. Un nombre de priorité élevé correspond à une priorité plus importante. Réalisez l'exercice dans une approche avec préemption et sans préemption. Calculez ensuite le temps de rotation de chaque processus.

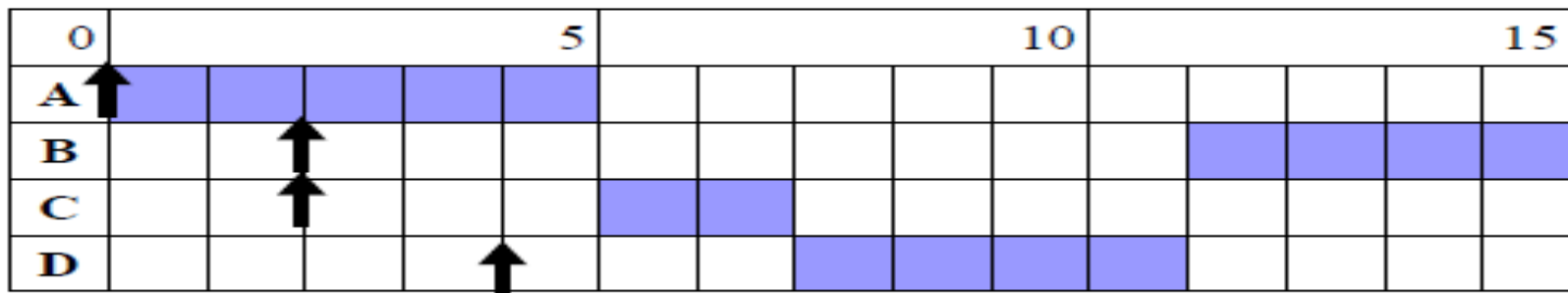
Processus	Date d'arrivée	Temps de traitement	Priorité
A	0	5	4
B	2	4	2
C	2	2	6
D	4	4	3

Pour chaque cas étudié, calculez :

- Temps de rotation de chaque processus et le temps de rotation moyen
- Temps d'attente de chaque processus et le temps d'attente moyen
- Rendement (*throughput*)

# Exemple

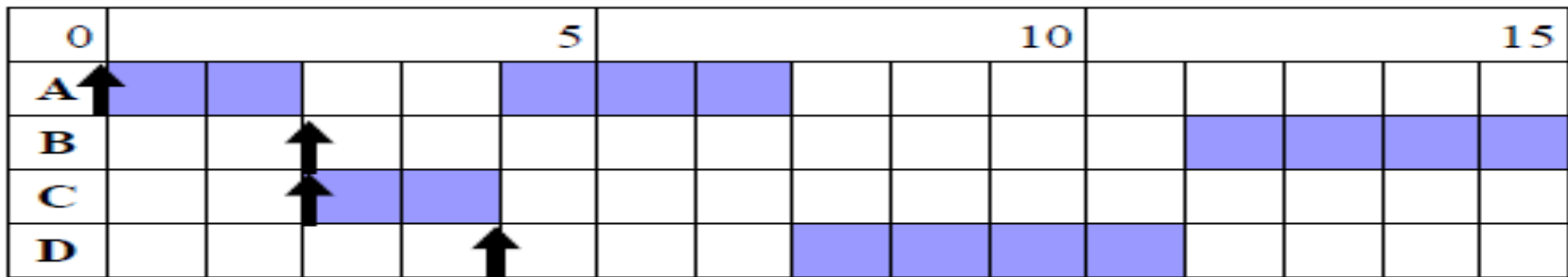
## Sans préemption



	Temps de rotation	Temps d'attente	Rendement
A	5	0	1
B	13	9	0,31
C	5	3	0,4
D	7	3	0,57
Moy	7,5	3,75	0,57

# Exemple

## Avec préemption



	Temps de rotation	Temps d'attente	Rendement
A	7	2	0,71
B	13	9	0,31
C	2	0	1
D	7	3	0,57
Moy	7,25	3,5	0,65

# Gestion des processus sous linux

---

Processus = suite plus ou moins longue d'instructions exécutables par un ordinateur (typiquement un programme)

Unix = système multi-tâches

-> Unix peut gérer plusieurs processus en même temps

Quelques commandes de gestion des processus :

- `top` et `ps` : permettent de visualiser les processus qui sont en exécution
- `kill` : permet de tuer un processus

Notion d'avant-plan et d'arrière-plan :

- avant-plan : on a pas la main dans le shell (par défaut quand on lance une commande)
- arrière-plan : on récupère la main dans le shell alors que le processus tourne
- pour lancer une commande en arrière-plan : `commande &`

---

Dans un terminal, on peut surveiller son système avec 2 outils : **top**

On pourra également tuer des processus en se positionnant dessus puis en lançant la commande kill (F9). Avec top, on utilise la commande **kill** avec le signal en argument : **-15** pour arrêter proprement le processus, **-9** pour le tuer.

Exemple pour tuer Firefox : **kill -9 4694**

# Les touches magiques

---

Le PC ne répond plus et vous vous apprêtez à arracher la prise de courant, appuyez longuement sur le bouton **Power, STOP** !

C'est une combinaison de touches pour envoyer des ordres directement au noyau de votre système. Les touches à appuyer sont **Alt + SysRq + une\_autre\_touche**

La touche SysRq est généralement la touche Impr\_Ecran sur les claviers. Cette fonctionnalité est activée par défaut dans Ubuntu, et cette activation se fait directement dans le noyau.

# Dans l'ordre

---

**s** pour synchronise les disques

**e** pour essaie de fermer les processus en envoyant SIGTERM (facultatif)

**i** pour tue tous les processus restant en envoyant SIGKILL (facultatif)

**u** pour démonte tous les disques

**b** pour redémarre

Touche SystRq est la même touche que Imprim\_ÉcranInformations

Il faut donc enchaîner Alt + Impr\_Ecran + s, Alt + Impr\_Ecran + e... et ainsi de suite ; attention, il est important d'attendre plusieurs secondes entre les différentes étapes !

**Savoir Eteindre Intégralement Ubuntu Brusquement**



Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

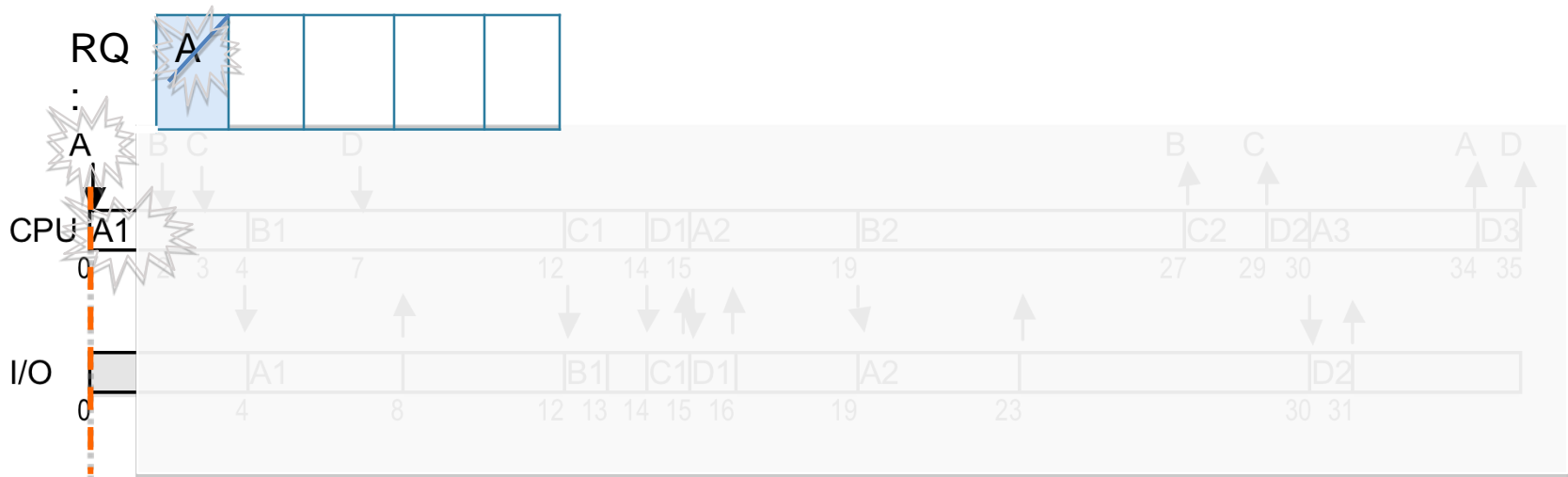
# Exercise

---

**PREMIER ARRIVÉ**  
**PREMIER SERVI**

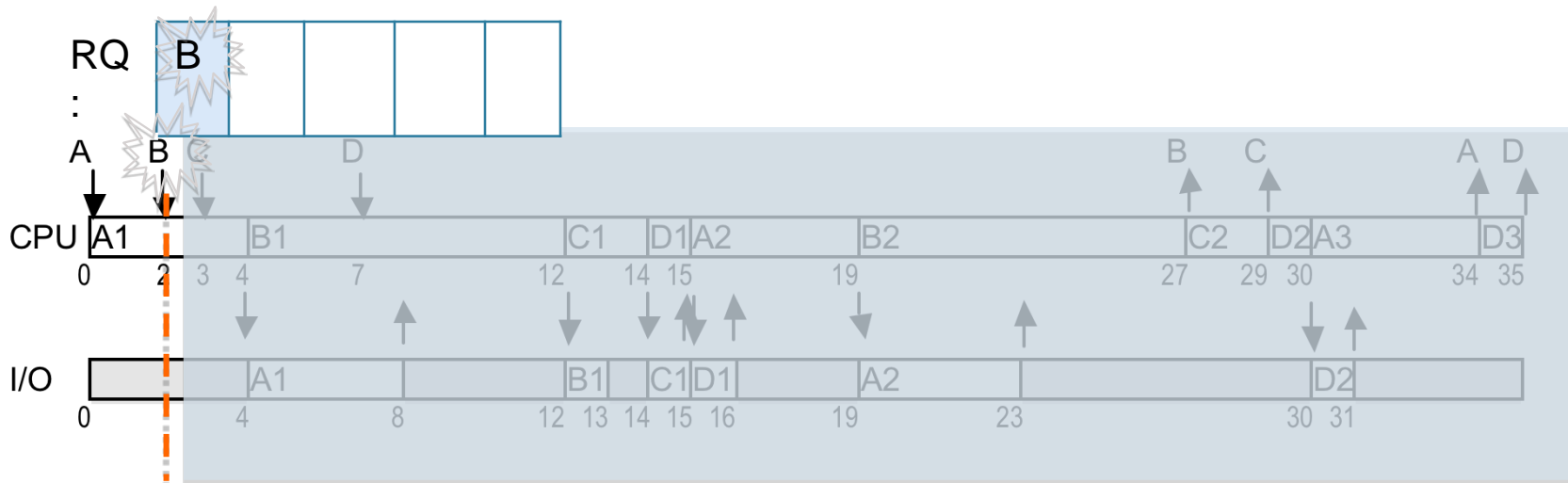
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



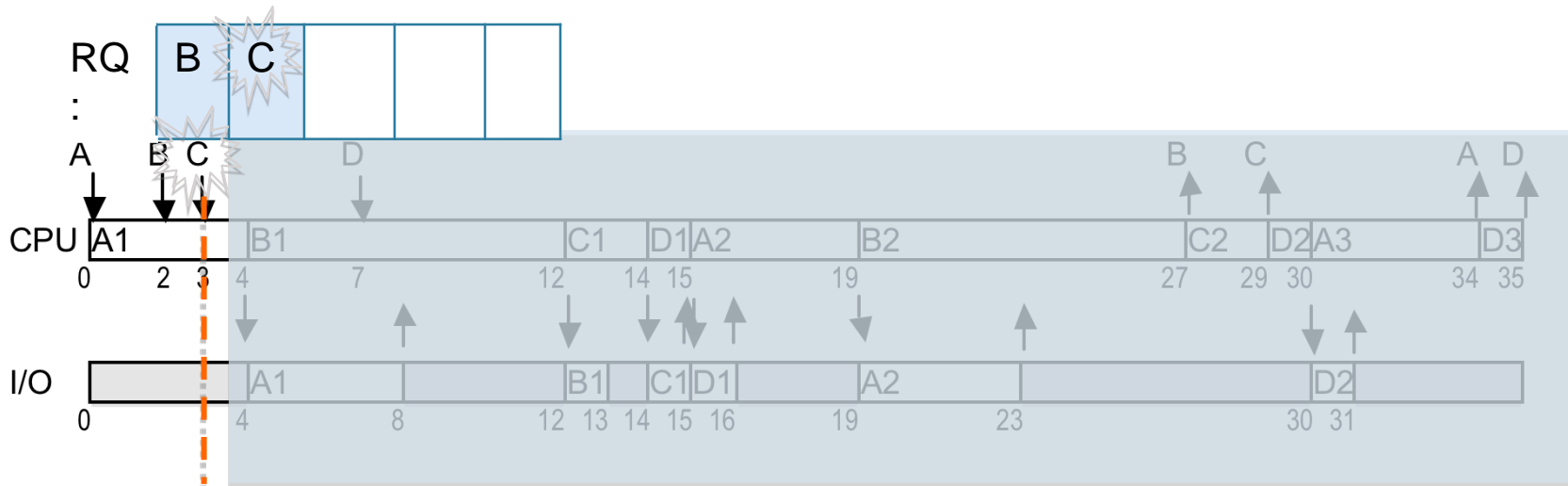
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



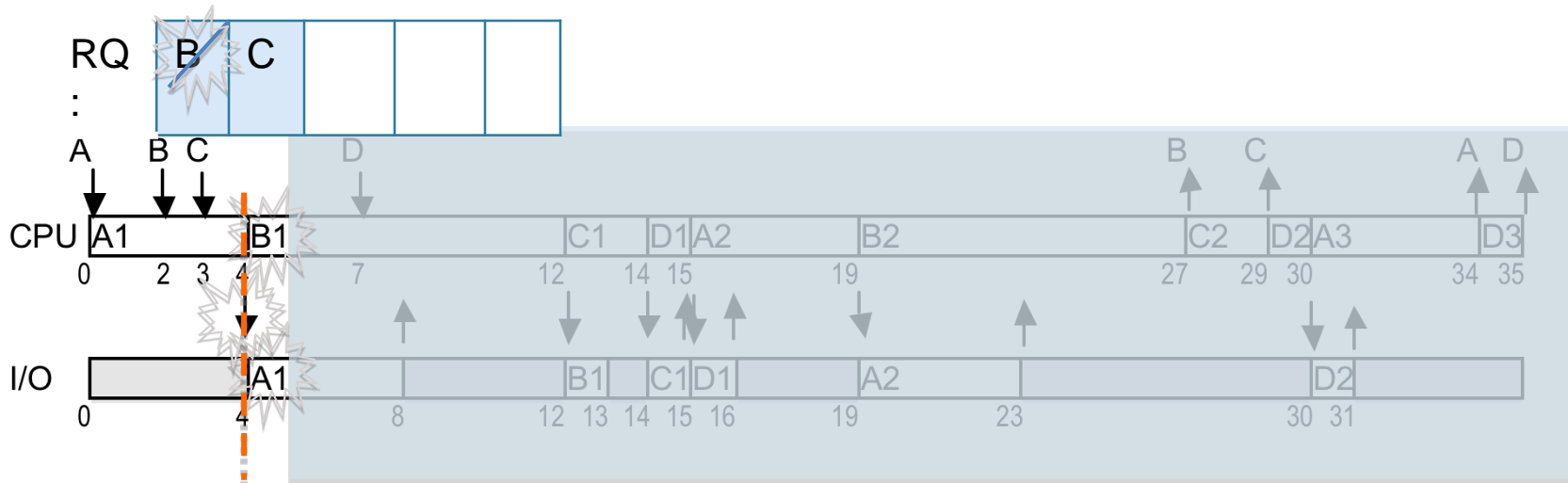
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



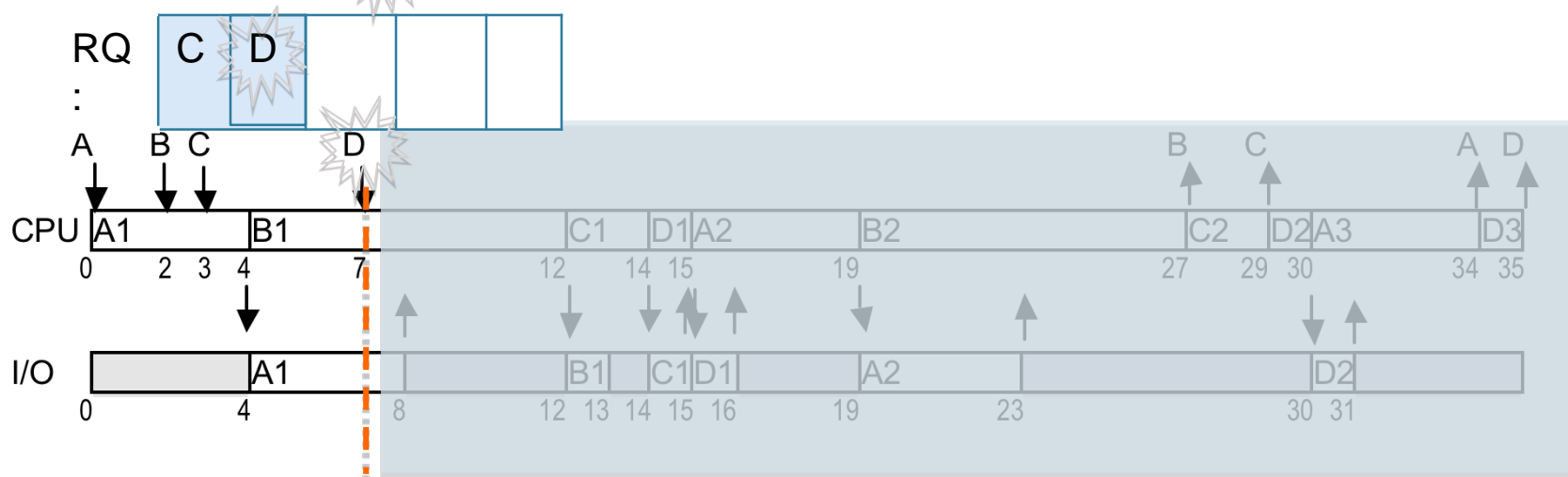
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



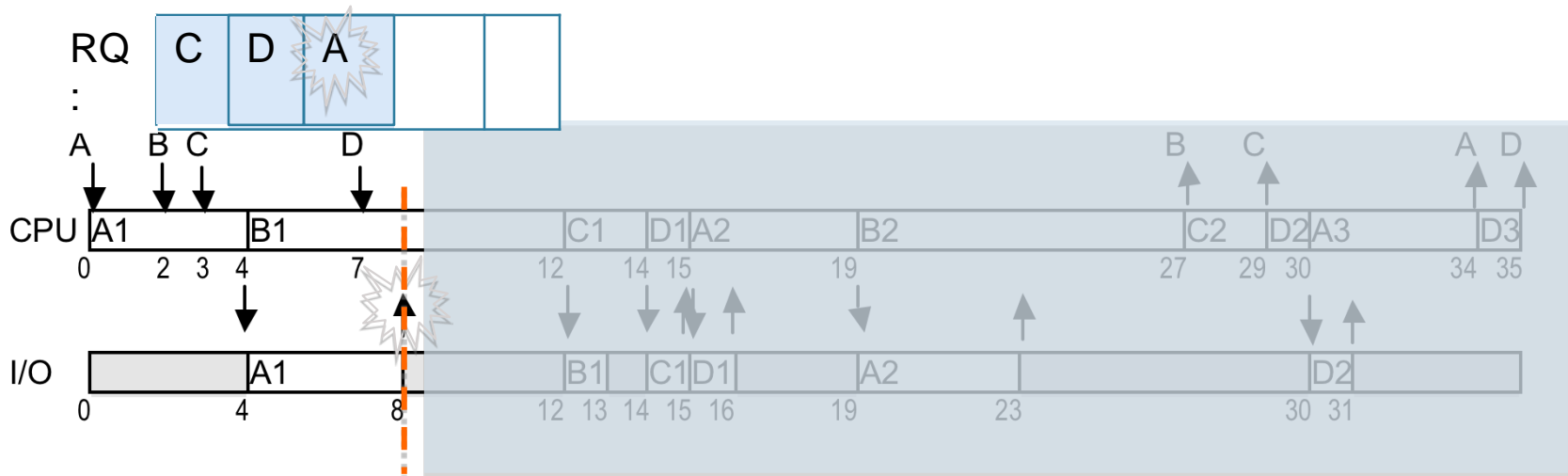
### 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.1 First-Come-First-Served (FCFS)

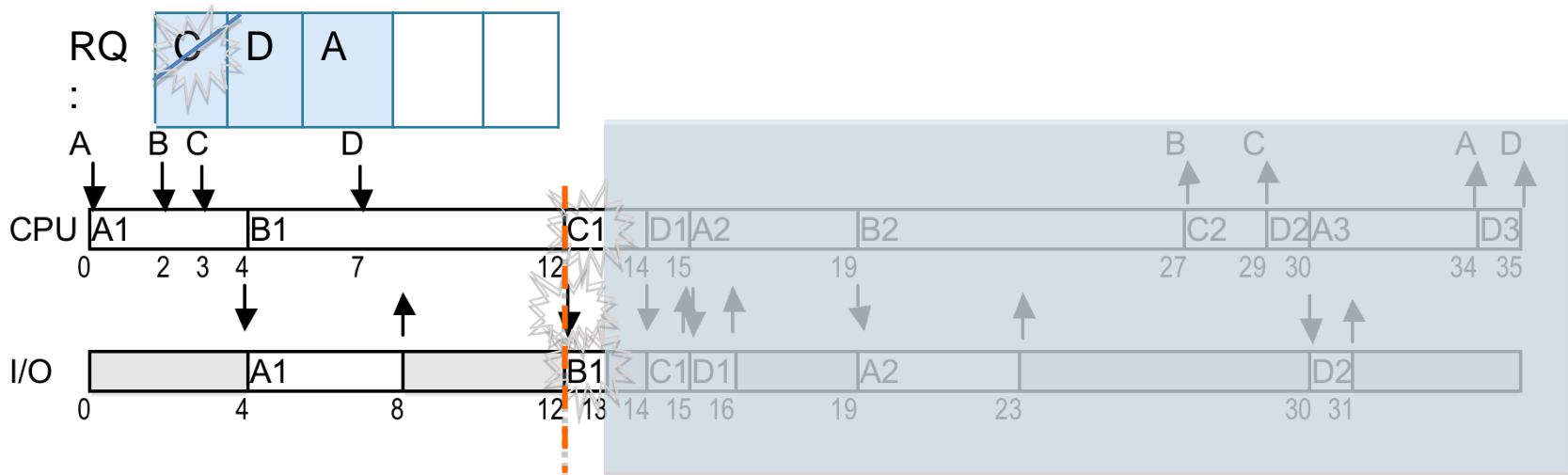
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1





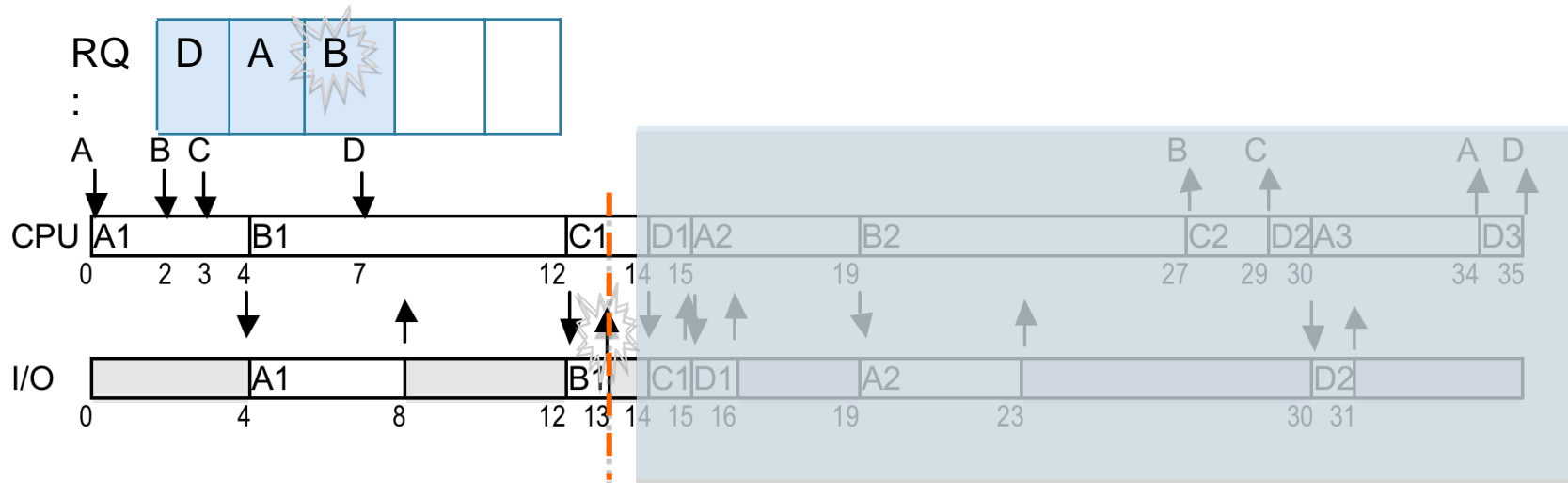
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



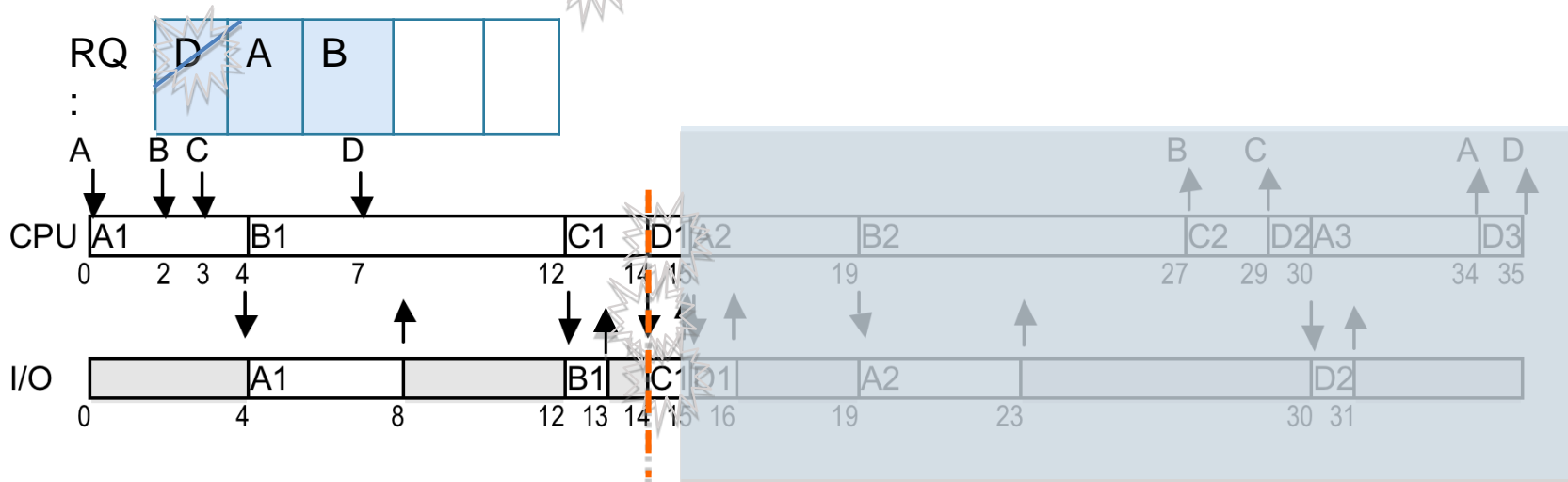
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



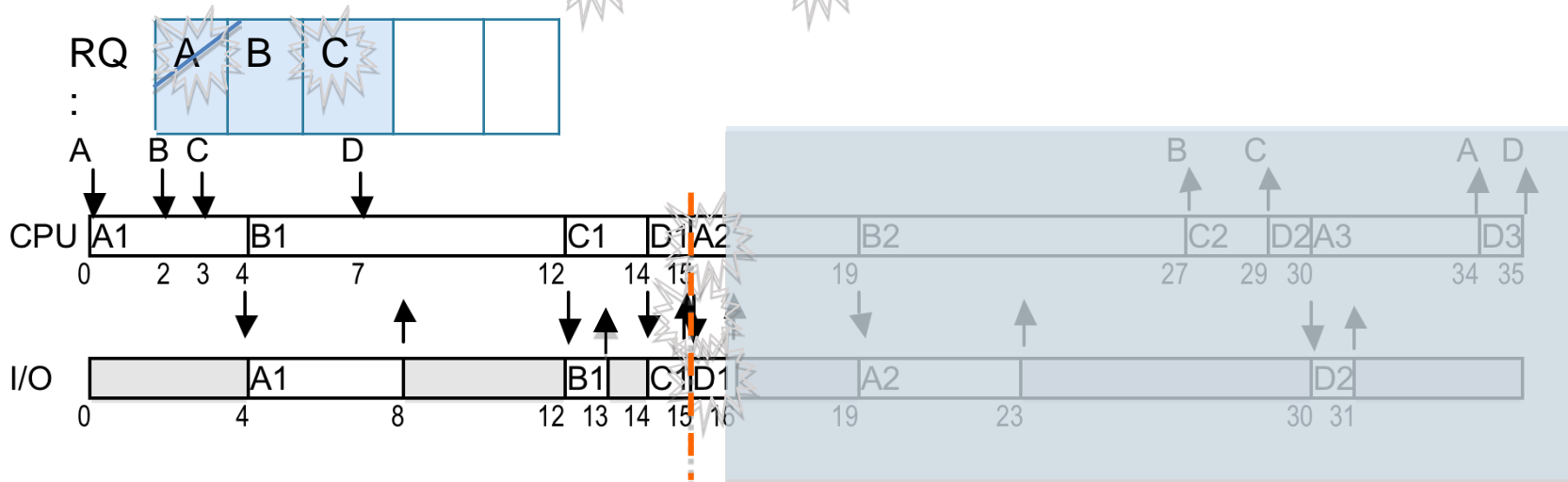
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



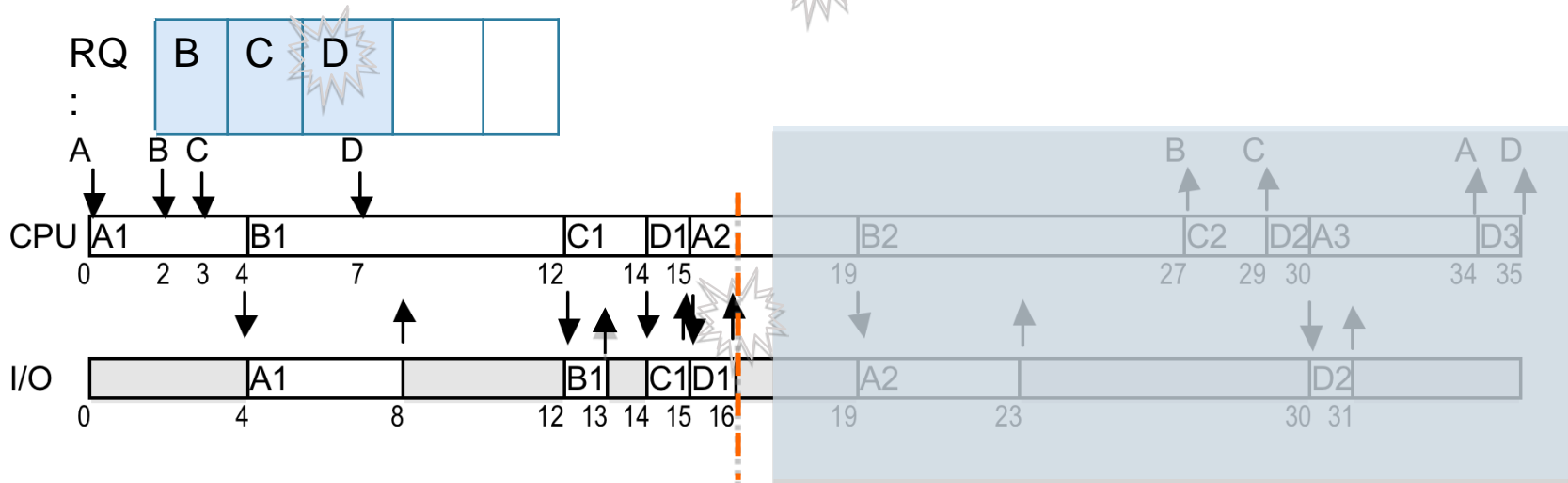
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



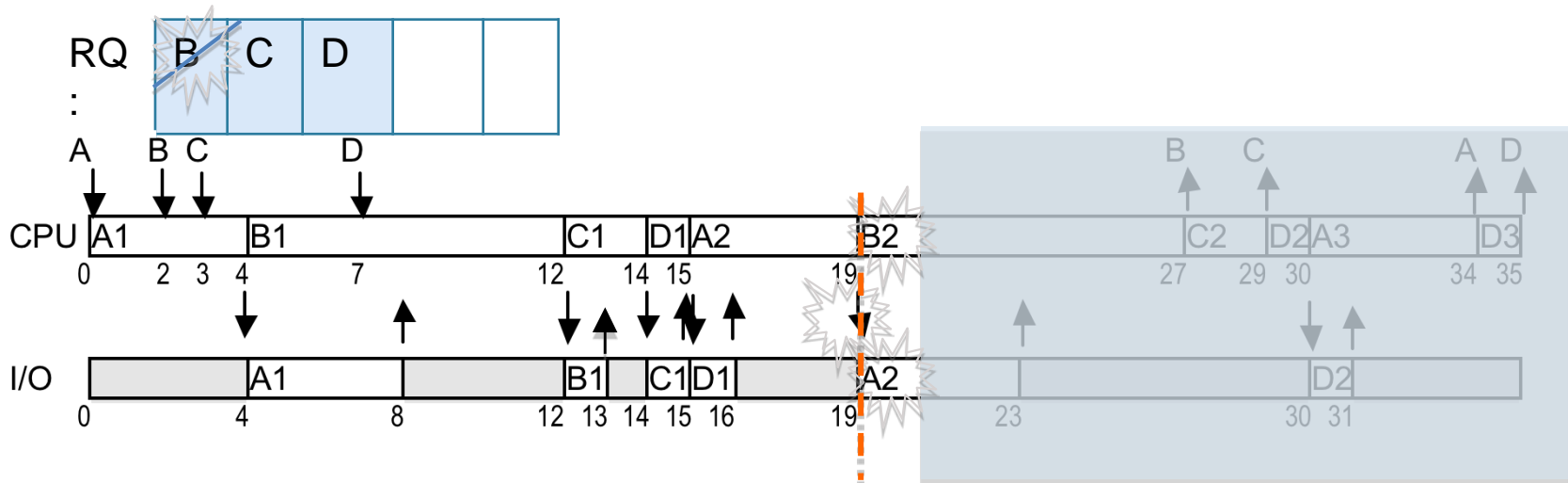
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



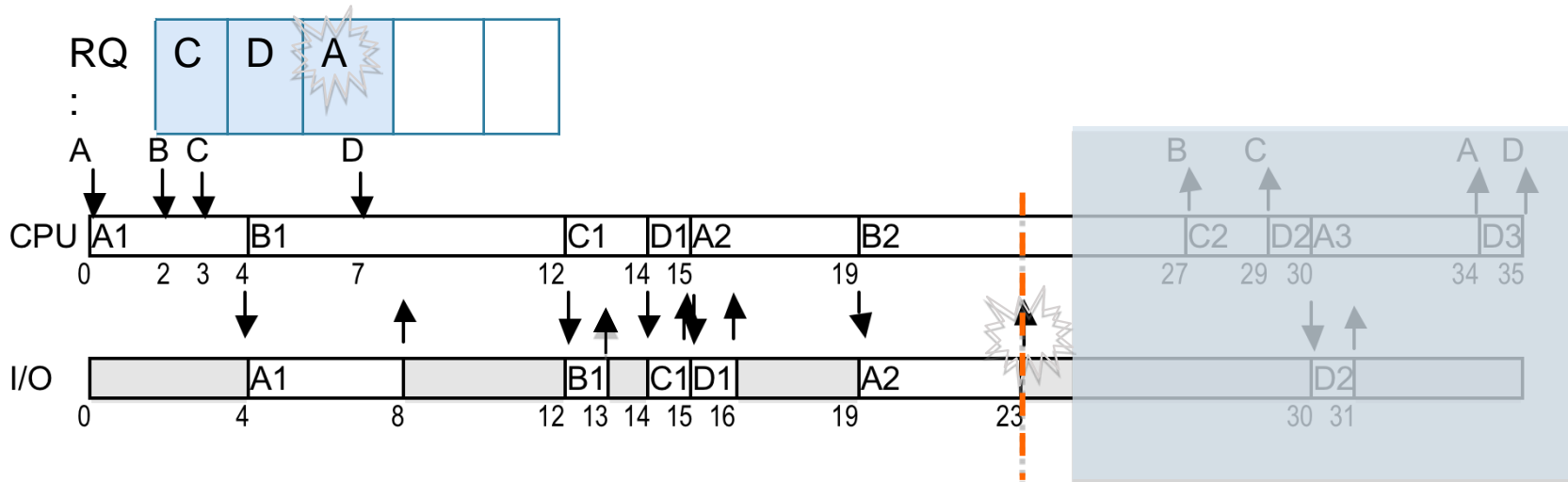
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	5	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



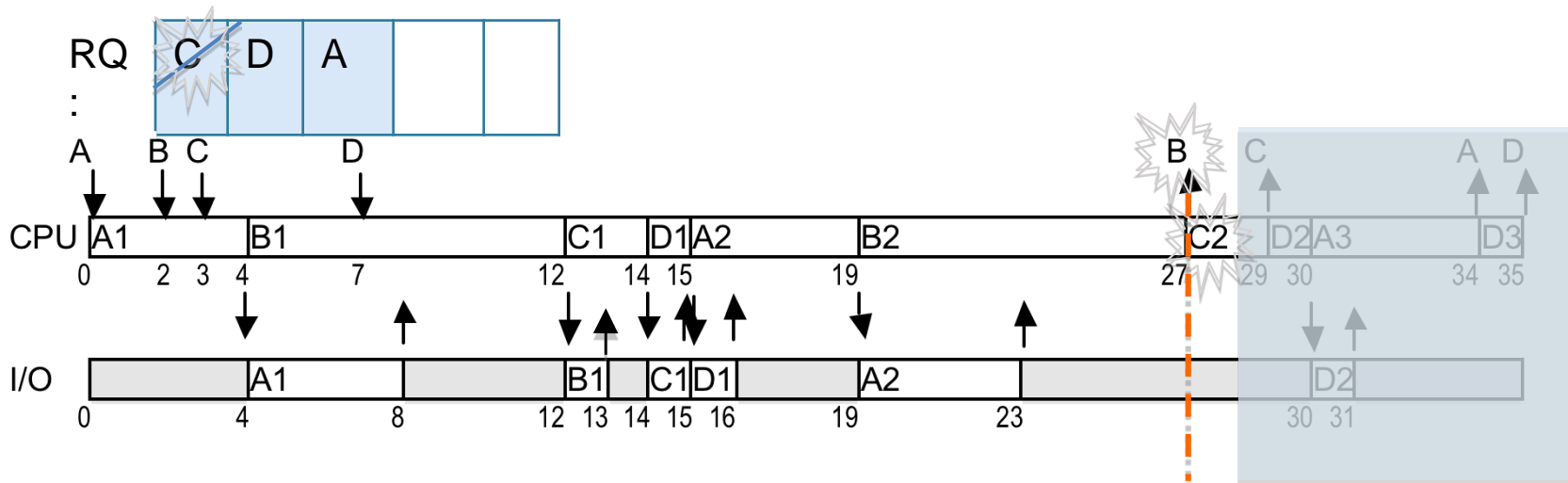
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.1 First-Come-First-Served (FCFS)

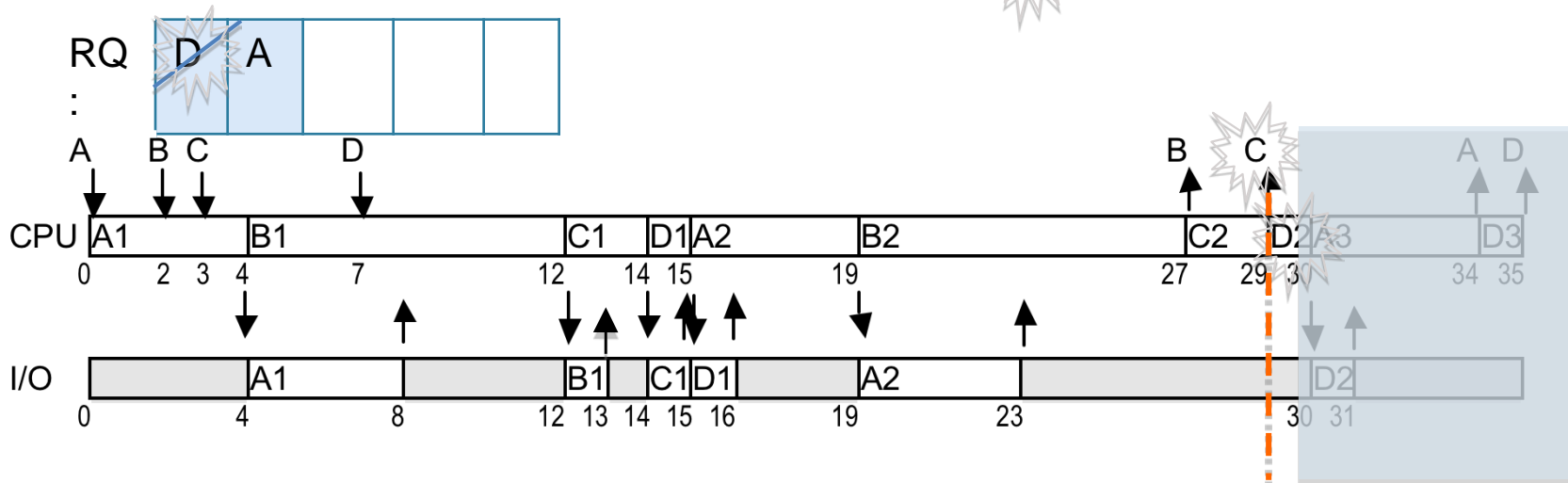
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	4	-	-
D	7	1	1	1	1	1





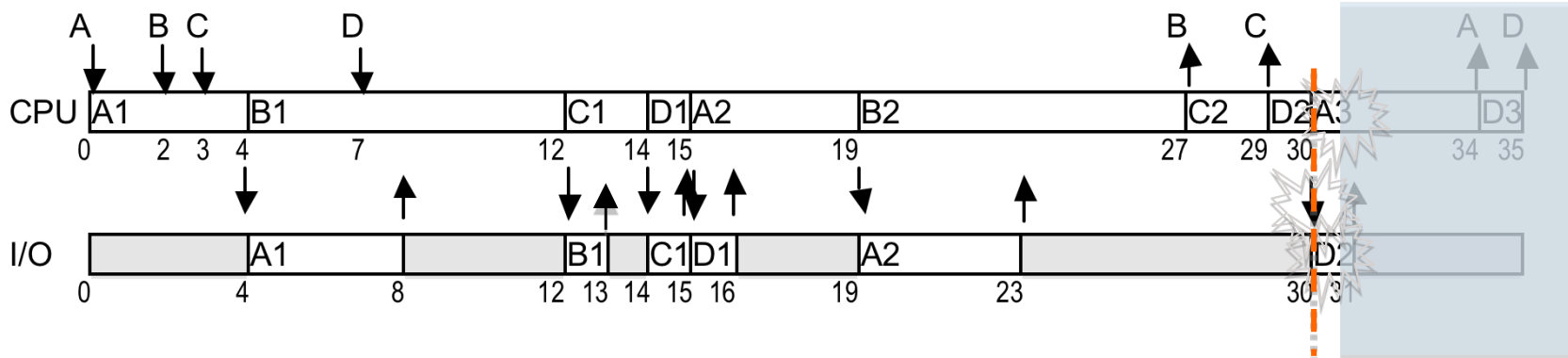
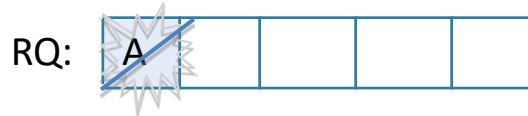
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.1 First-Come-First-Served (FCFS)

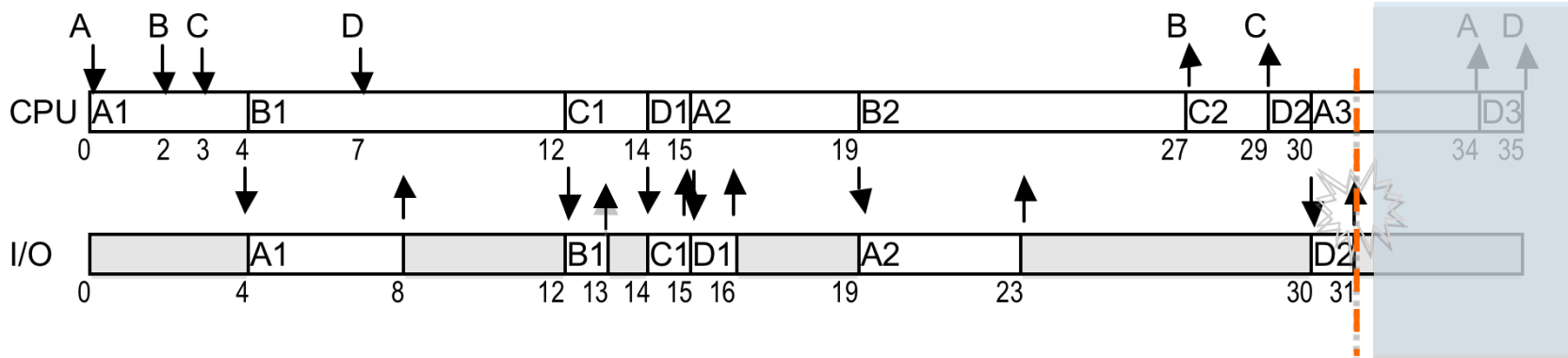
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

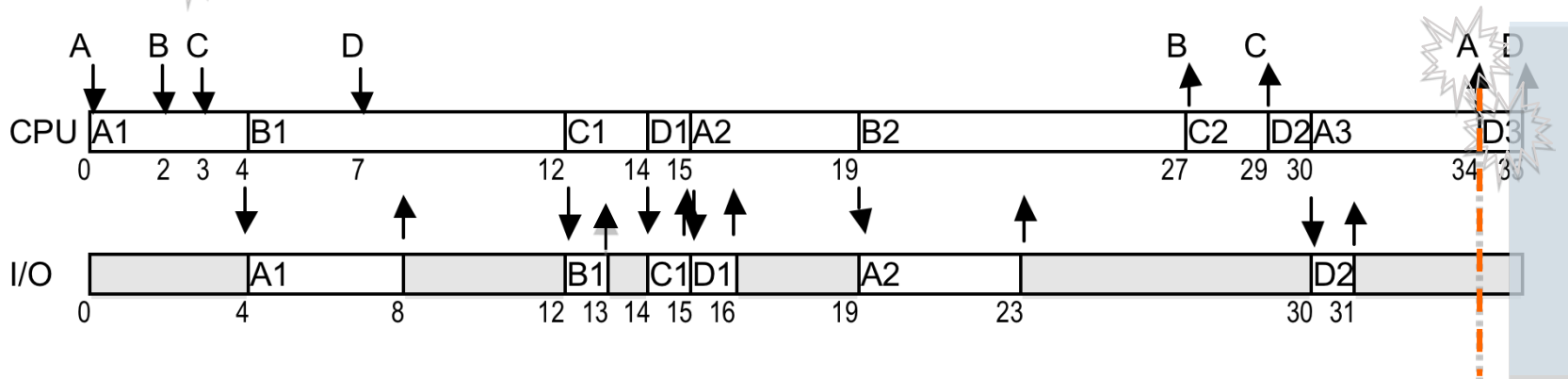
RQ: 



## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

RQ: 



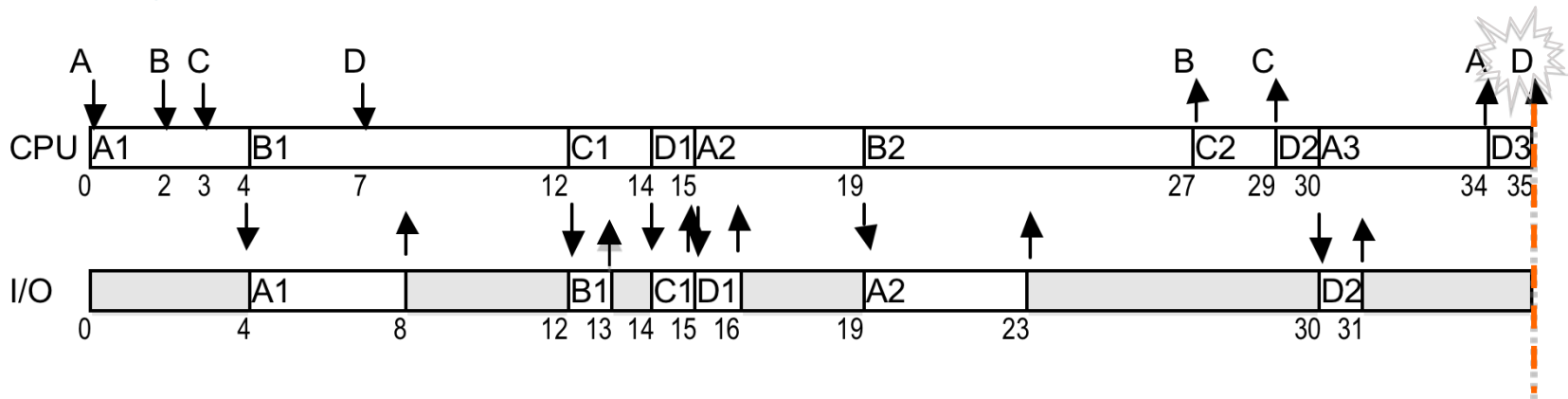
## 2.3.1 First-Come-First-Served (FCFS)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

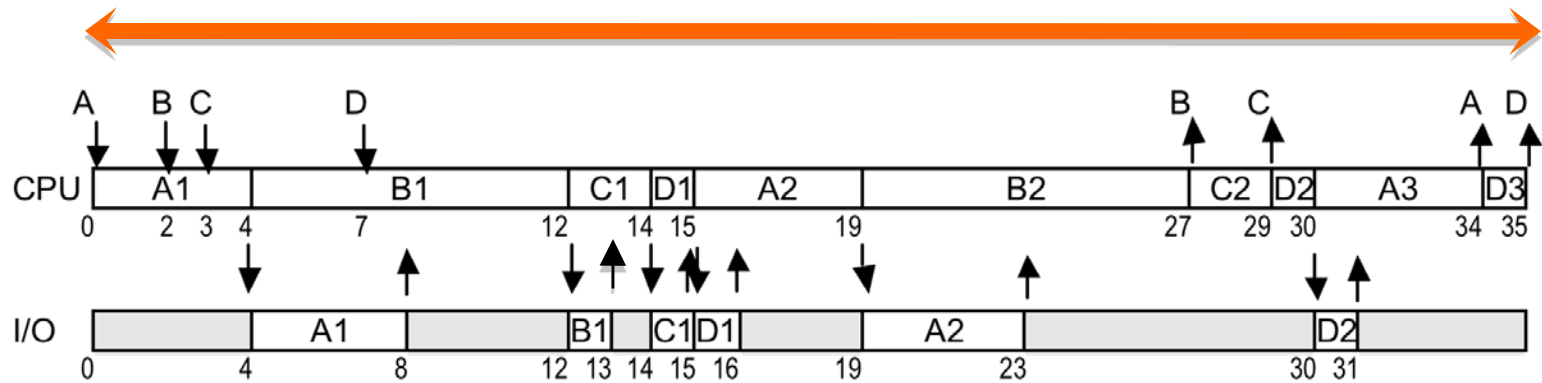


RQ:

--	--	--	--	--

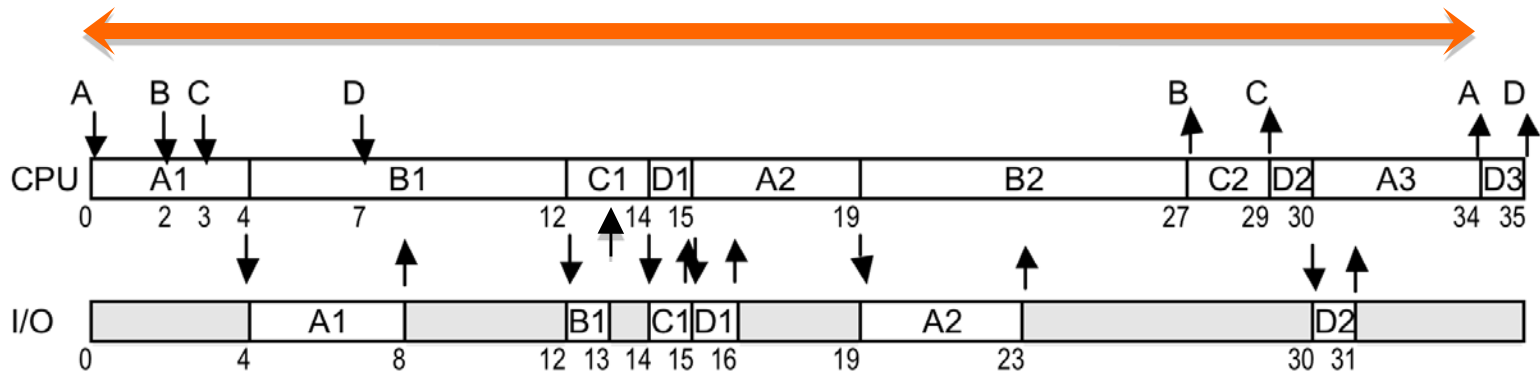


## 2.3.1 First-Come-First-Served (FCFS)



- Processor utilization =  $(35 / 35) * 100 = 100 \%$
- Throughput =  $4 / 35 = 0.11$

## 2.3.1 First-Come-First-Served (FCFS)



- Turn around time:

$$tat_A = 34 - 0 = 34$$

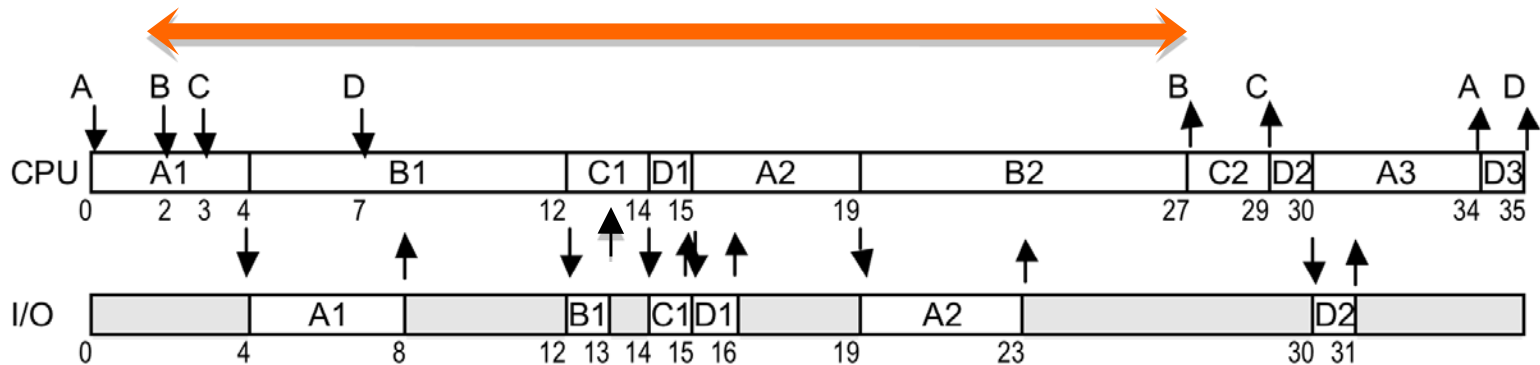
$$tat_B = 27 - 2 = 25$$

$$tat_C = 29 - 3 = 26$$

$$tat_D = 35 - 7 = 28$$

$$tat_{AVG} = (34 + 25 + 26 + 28) / 4 = 28.25$$

## 2.3.1 First-Come-First-Served (FCFS)



- Turn around time:

$$tat_A = 34 - 0 = 34$$

$$tat_B = 27 - 2 = 25$$

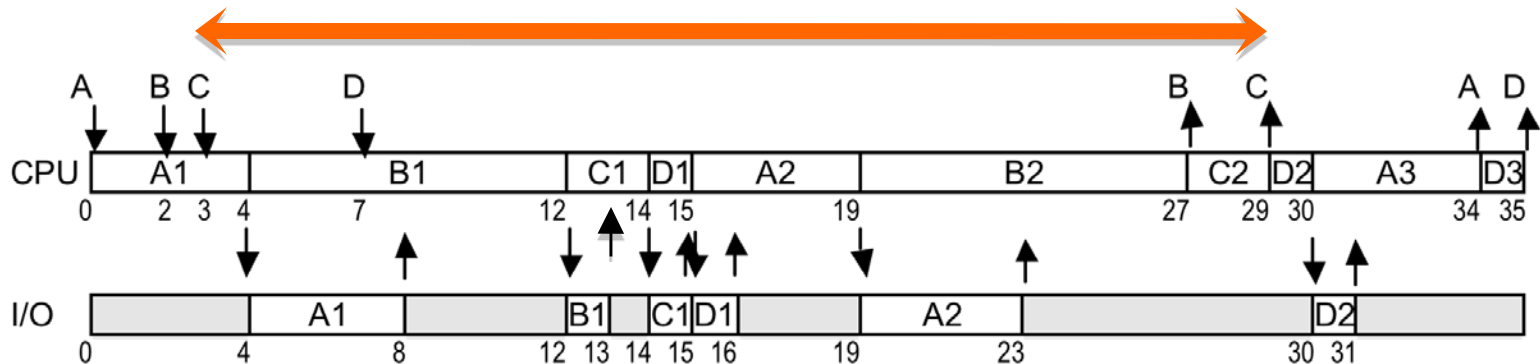
$$tat_C = 29 - 3 = 26$$

$$tat_D = 35 - 7 = 28$$

$$tat_{AVG} = (34 + 25 + 26 + 28) / 4 = 28.25$$



## 2.3.1 First-Come-First-Served (FCFS)



- Turn around time:

$$tat_A = 34 - 0 = 34$$

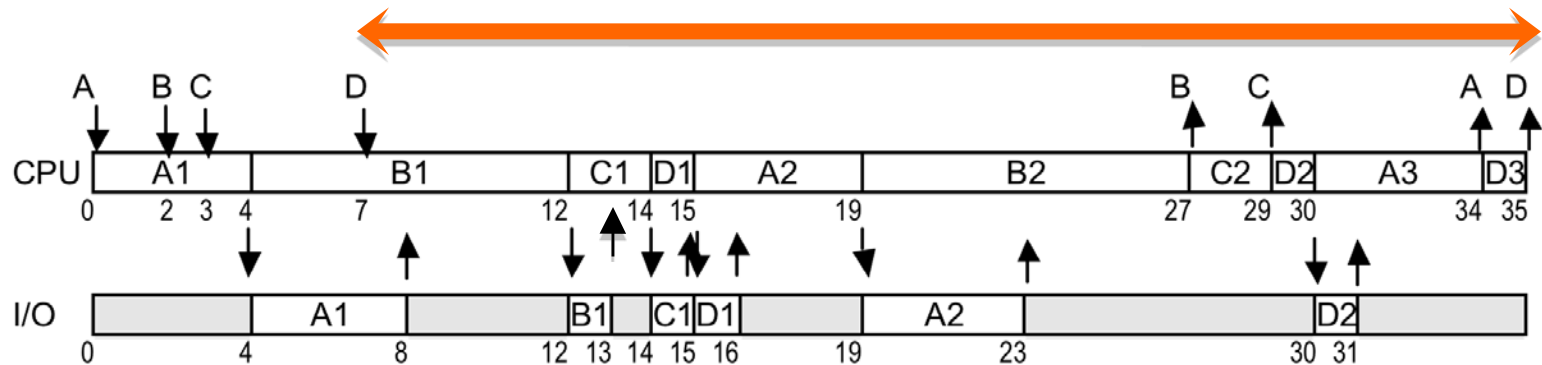
$$tat_B = 27 - 2 = 25$$

$$tat_C = 29 - 3 = 26$$

$$tat_D = 35 - 7 = 28$$

$$tat_{AVG} = (34 + 25 + 26 + 28) / 4 = 28.25$$

## 2.3.1 First-Come-First-Served (FCFS)



- Turn around time:

$$tat_A = 34 - 0 = 34$$

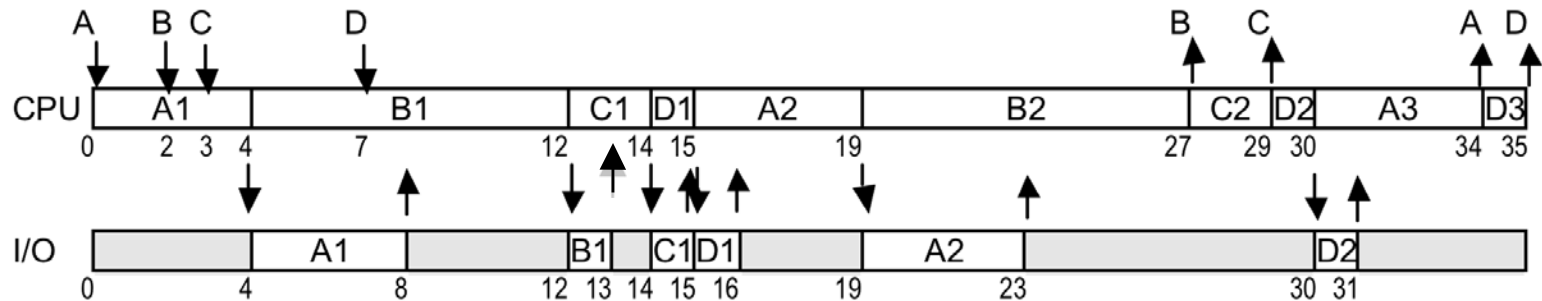
$$tat_B = 27 - 2 = 25$$

$$tat_C = 29 - 3 = 26$$

$$tat_D = 35 - 7 = 28$$

$$tat_{AVG} = (34 + 25 + 26 + 28) / 4 = 28.25$$

## 2.3.1 First-Come-First-Served (FCFS)



- Turn around time:

$$\text{tat}_A = 34 - 0 = 34$$

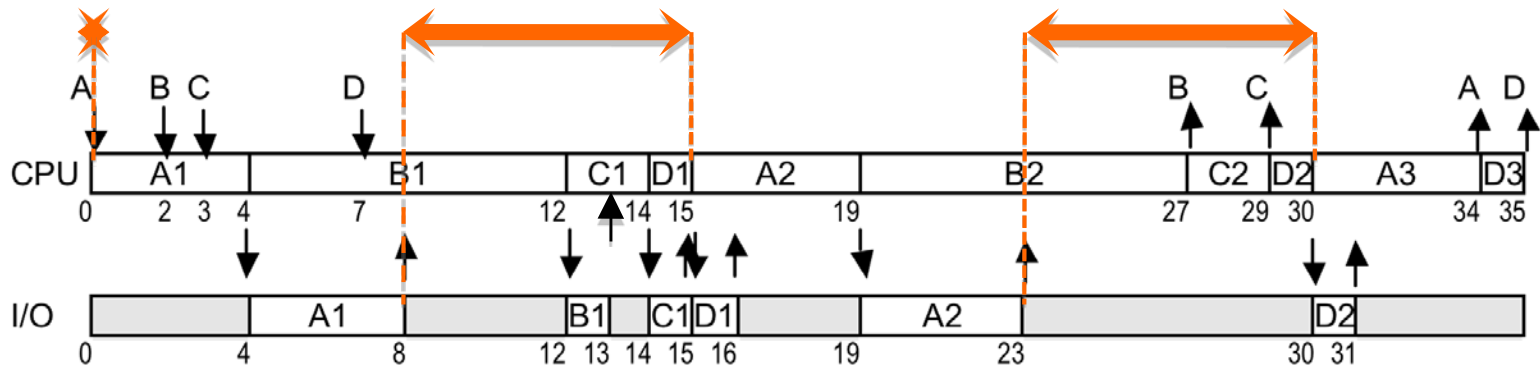
$$\text{tat}_B = 27 - 2 = 25$$

$$\text{tat}_C = 29 - 3 = 26$$

$$\text{tat}_D = 35 - 7 = 28$$

$$\text{tat}_{\text{AVG}} = (34 + 25 + 26 + 28) / 4 = 28.25$$

## 2.3.1 First-Come-First-Served (FCFS)



- Waiting time:

$$wt_A = (0 - 0) + (15 - 8) + (30 - 23) = 14$$

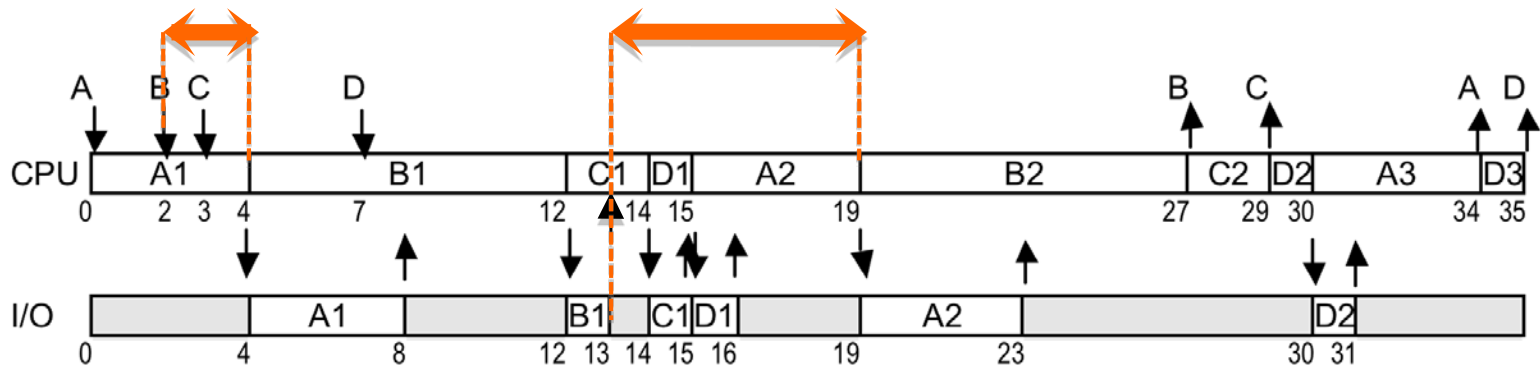
$$wt_B = (4 - 2) + (19 - 13) = 8$$

$$wt_C = (12 - 3) + (27 - 15) = 21$$

$$wt_D = (14 - 7) + (29 - 16) + (34 - 31) = 23$$

$$wt_{AVG} = (14 + 12 + 21 + 23) / 4 = 16.5$$

## 2.3.1 First-Come-First-Served (FCFS)



- Waiting time:

$$wt_A = (0 - 0) + (15 - 8) + (30 - 23) = 14$$

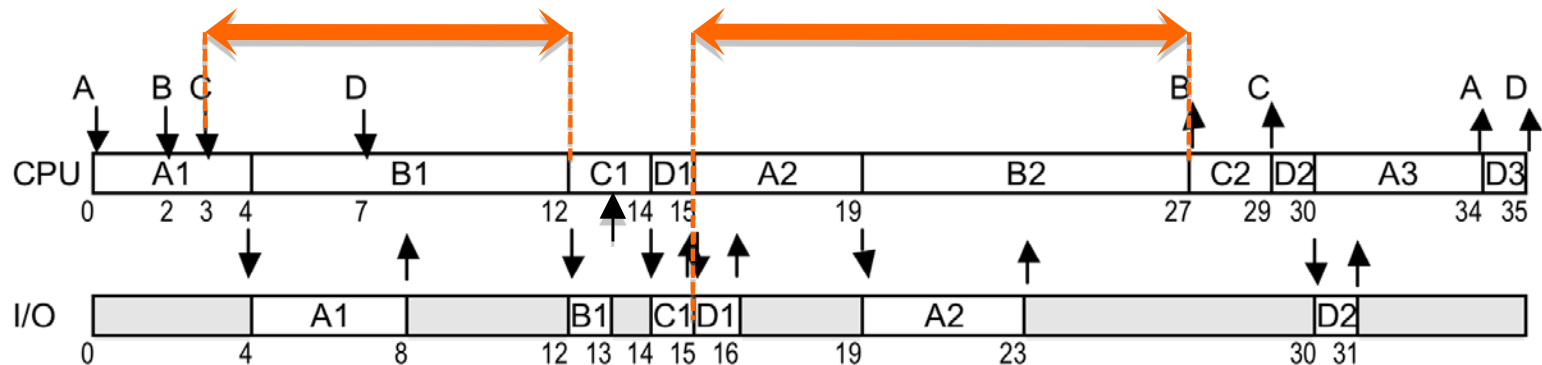
$$wt_B = (4 - 2) + (19 - 13) = 8$$

$$wt_C = (12 - 3) + (27 - 15) = 21$$

$$wt_D = (14 - 7) + (29 - 16) + (34 - 31) = 23$$

$$wt_{AVG} = (14 + 12 + 21 + 23) / 4 = 16.5$$

## 2.3.1 First-Come-First-Served (FCFS)



- Waiting time:

$$wt_A = (0 - 0) + (15 - 8) + (30 - 23) = 14$$

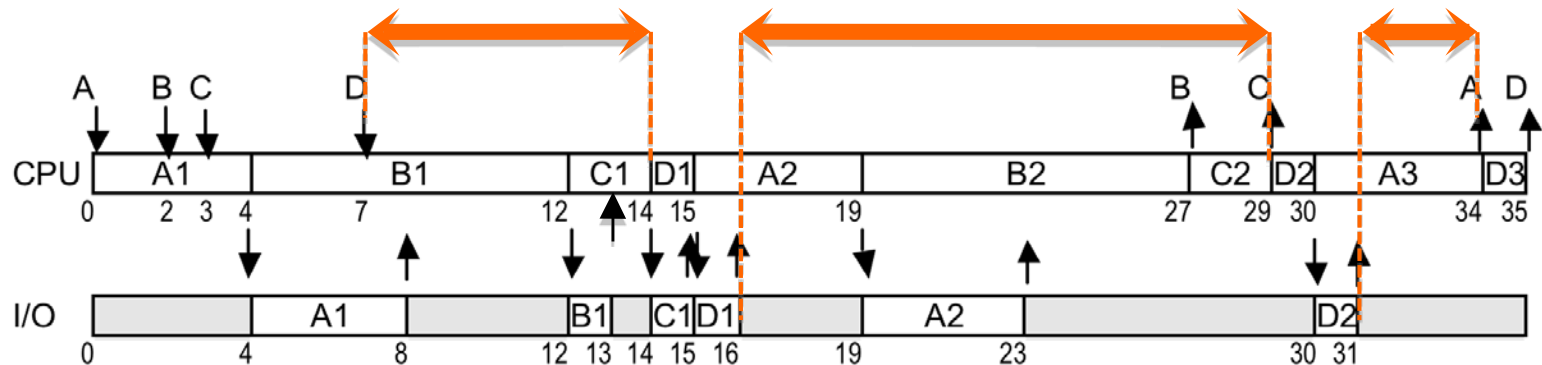
$$wt_B = (4 - 2) + (19 - 13) = 8$$

$$wt_C = (12 - 3) + (27 - 15) = 21$$

$$wt_D = (14 - 7) + (29 - 16) + (34 - 31) = 23$$

$$wt_{AVG} = (14 + 12 + 21 + 23) / 4 = 16.5$$

## 2.3.1 First-Come-First-Served (FCFS)



- Waiting time:

$$wt_A = (0 - 0) + (15 - 8) + (30 - 23) = 14$$

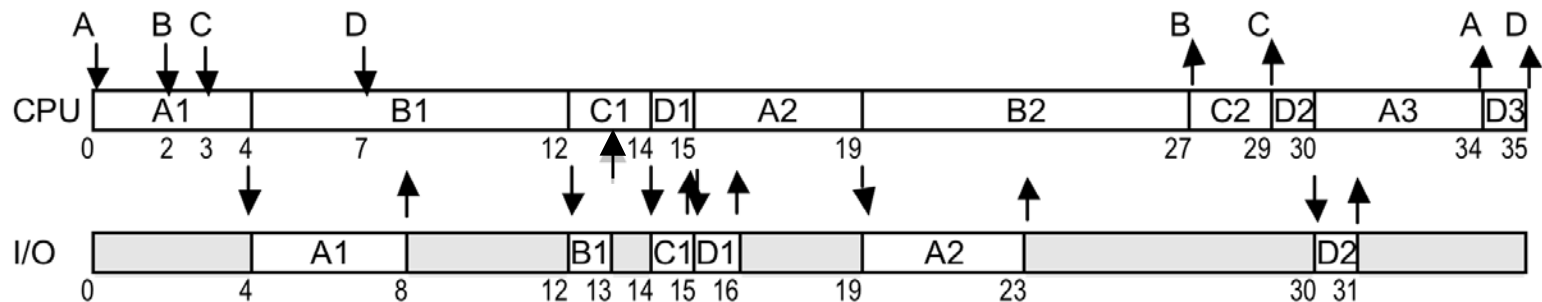
$$wt_B = (4 - 2) + (19 - 13) = 8$$

$$wt_C = (12 - 3) + (27 - 15) = 21$$

$$wt_D = (14 - 7) + (29 - 16) + (34 - 31) = 23$$

$$wt_{AVG} = (14 + 12 + 21 + 23) / 4 = 16.5$$

## 2.3.1 First-Come-First-Served (FCFS)



- Waiting time:

$$wt_A = (0 - 0) + (15 - 8) + (30 - 23) = 14$$

$$wt_B = (4 - 2) + (19 - 13) = 8$$

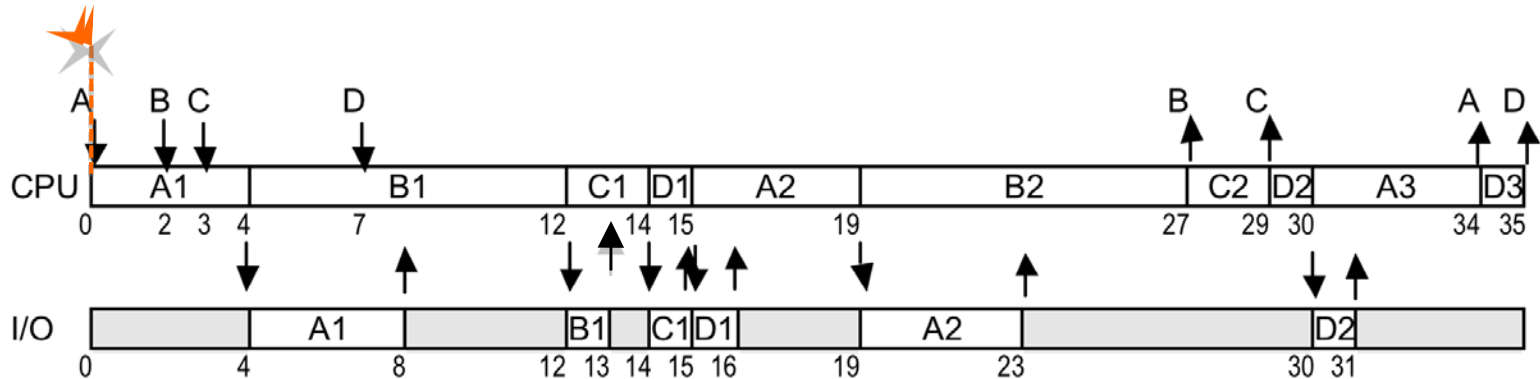
$$wt_C = (12 - 3) + (27 - 15) = 21$$

$$wt_D = (14 - 7) + (29 - 16) + (34 - 31) = 23$$

$$wt_{AVG} = (14 + 12 + 21 + 23) / 4 = 16.5$$



## 2.3.1 First-Come-First-Served (FCFS)



- Response time:

$$rt_A = 0 - 0 = 0$$

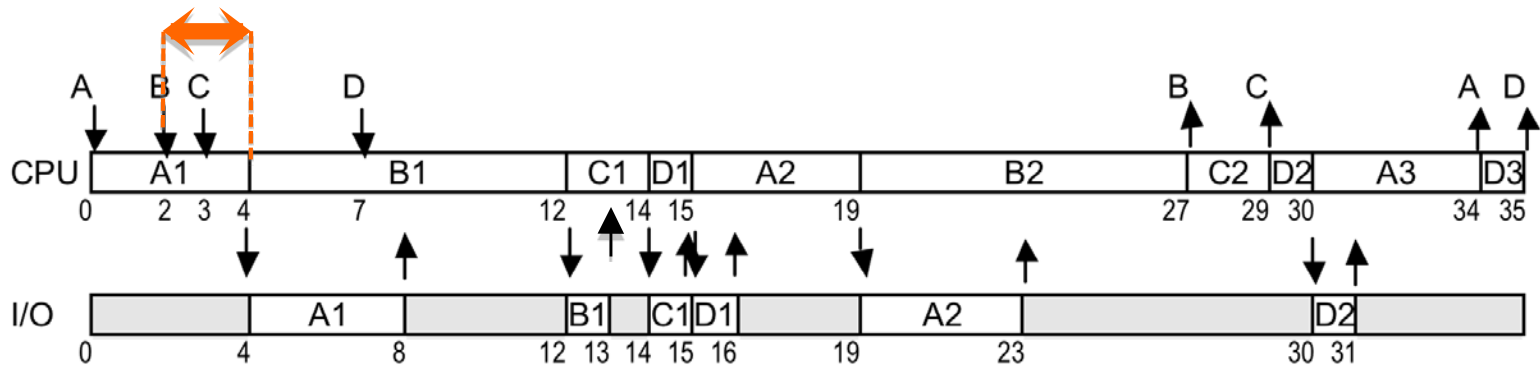
$$rt_B = 4 - 2 = 2$$

$$rt_C = 12 - 3 = 9$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 2 + 9 + 7) / 4 = 4.5$$

## 2.3.1 First-Come-First-Served (FCFS)



- Response time:

$$rt_A = 0 - 0 = 0$$

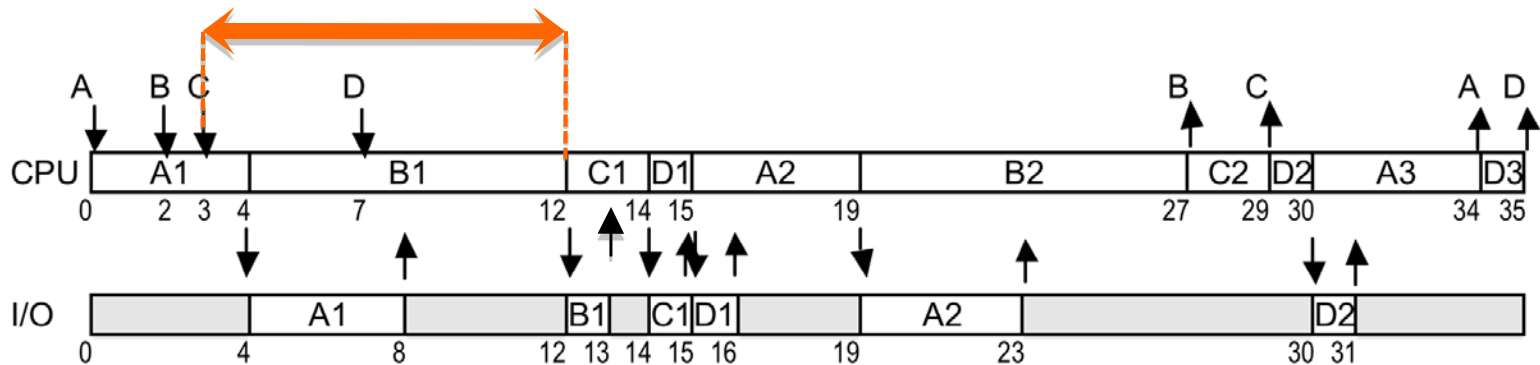
$$rt_B = 4 - 2 = 2$$

$$rt_C = 12 - 3 = 9$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 2 + 9 + 7) / 4 = 4.5$$

## 2.3.1 First-Come-First-Served (FCFS)



- Response time:

$$rt_A = 0 - 0 = 0$$

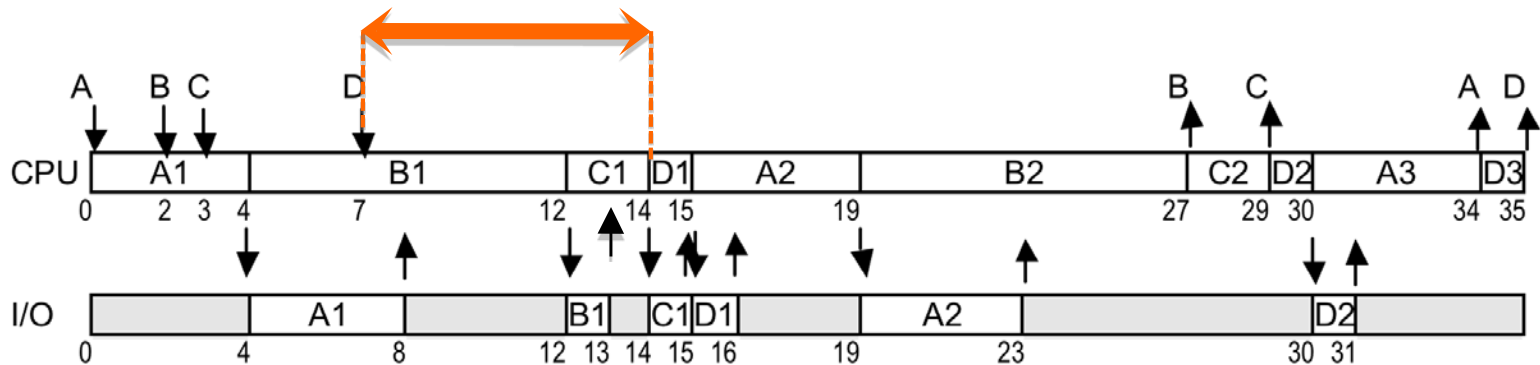
$$rt_B = 4 - 2 = 2$$

$$rt_C = 12 - 3 = 9$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 2 + 9 + 7) / 4 = 4.5$$

## 2.3.1 First-Come-First-Served (FCFS)



- Response time:

$$rt_A = 0 - 0 = 0$$

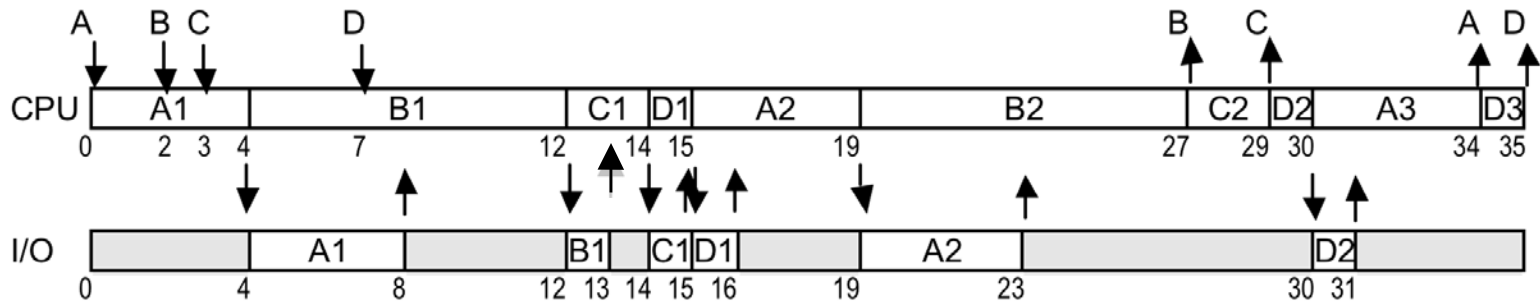
$$rt_B = 4 - 2 = 2$$

$$rt_C = 12 - 3 = 9$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 2 + 9 + 7) / 4 = 4.5$$

## 2.3.1 First-Come-First-Served (FCFS)



- Response time:

$$rt_A = 0 - 0 = 0$$

$$rt_B = 4 - 2 = 2$$

$$rt_C = 12 - 3 = 9$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 2 + 9 + 7) / 4 = 4.5$$

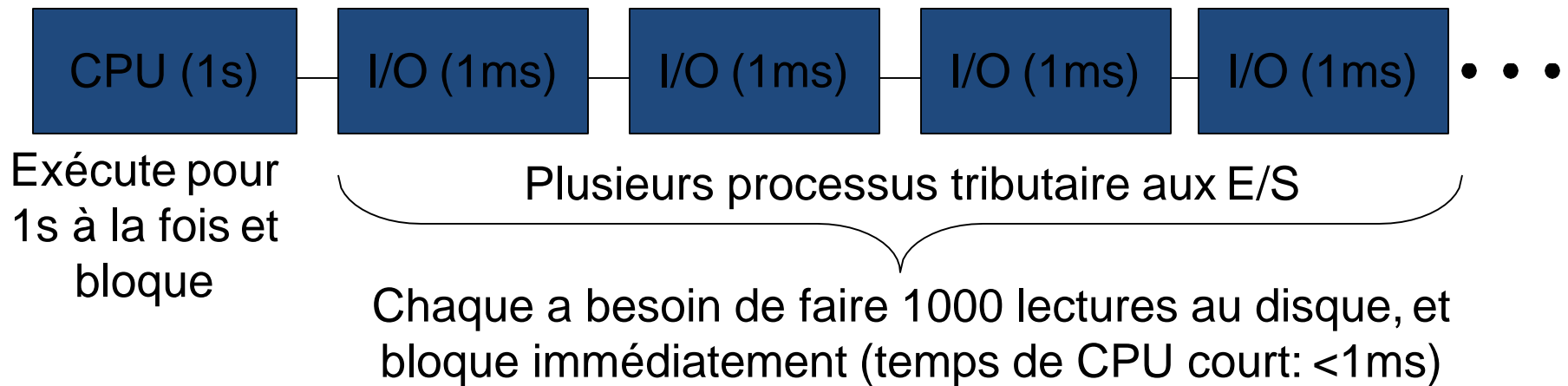
# Inconvénients FIFO

---

- défavorise les entités ayant besoin d'utiliser la ressource un court laps de temps
- Le temps d'attente n'est pas proportionnel au temps d'utilisation
  - ⇒ pas équitable,
  - ⇒ temps moyen de traitement élevé

# Inconvénients FIFO

- une grande perte de cycles de CPU



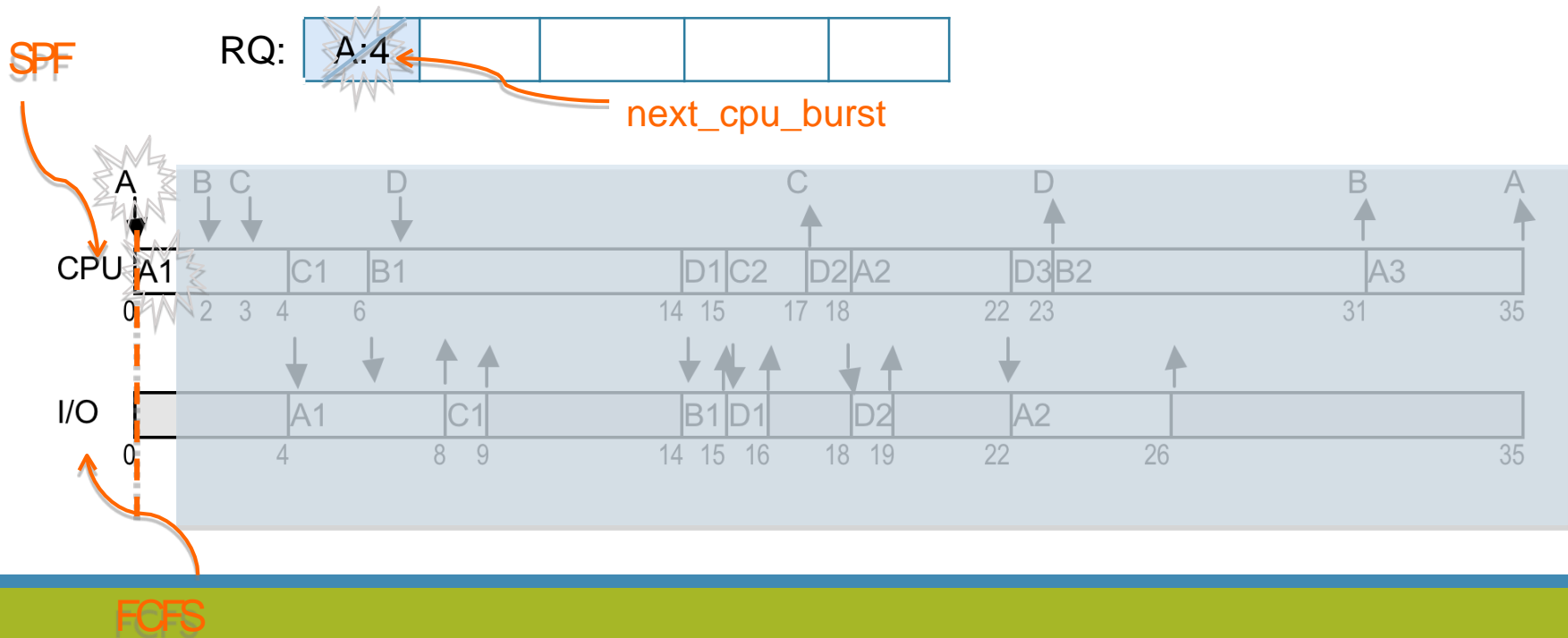
Si le processus d'E/S a le CPU une fois par seconde cela va prendre 1000 secondes pour que tout les processus complètent! Si le processus qui est tributaire du CPU donne le CPU tout les 10ms, tous les processus qui font des E/S finissent en un petit 10 secondes!

# Plut Court d'abord Shortest-Process- First (SPF)



## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

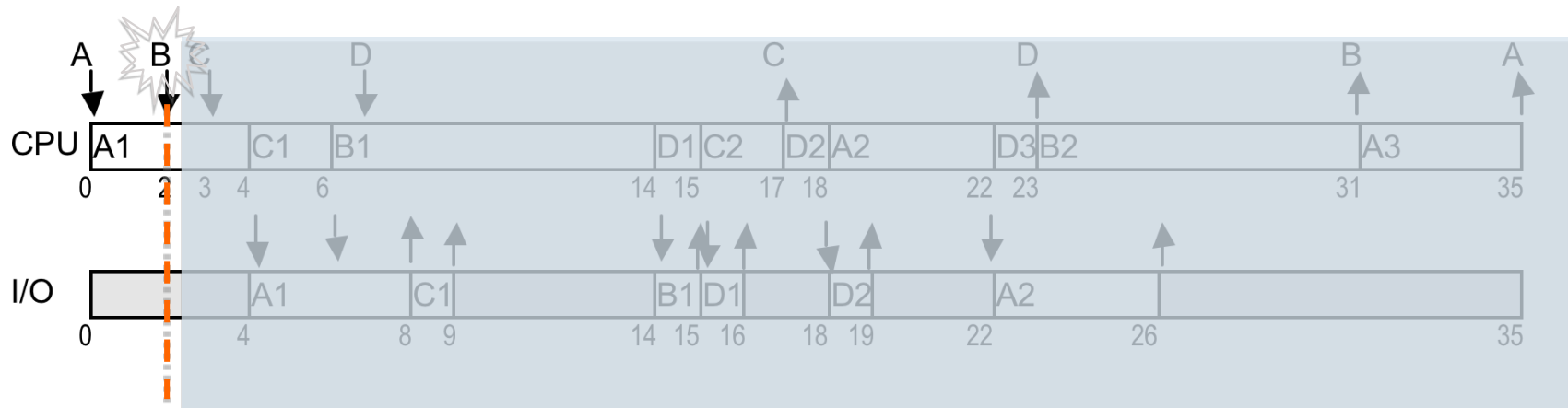


## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

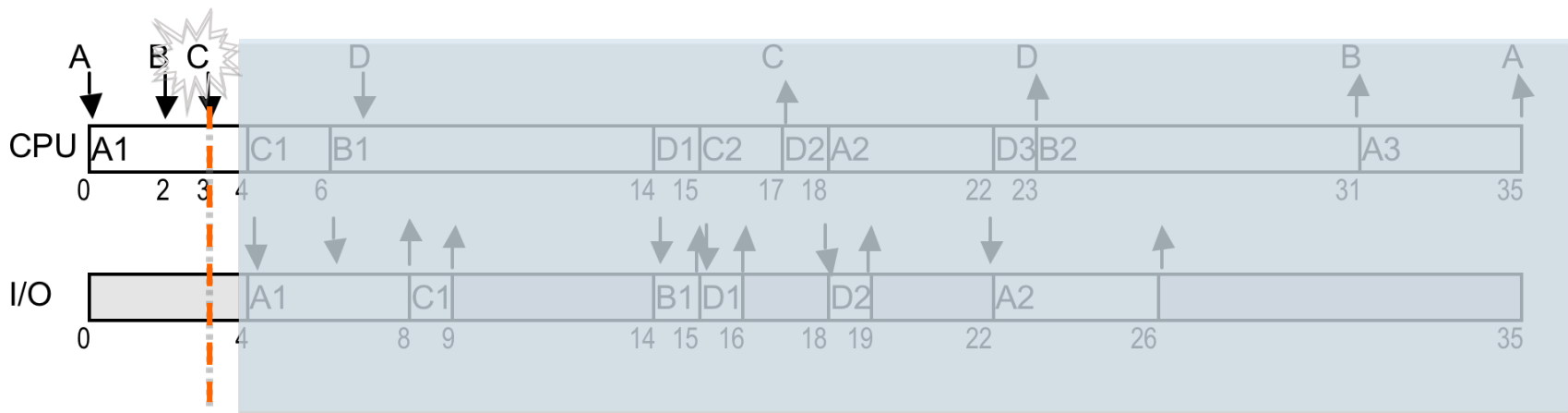


Proc\_id : next\_cpu\_burst



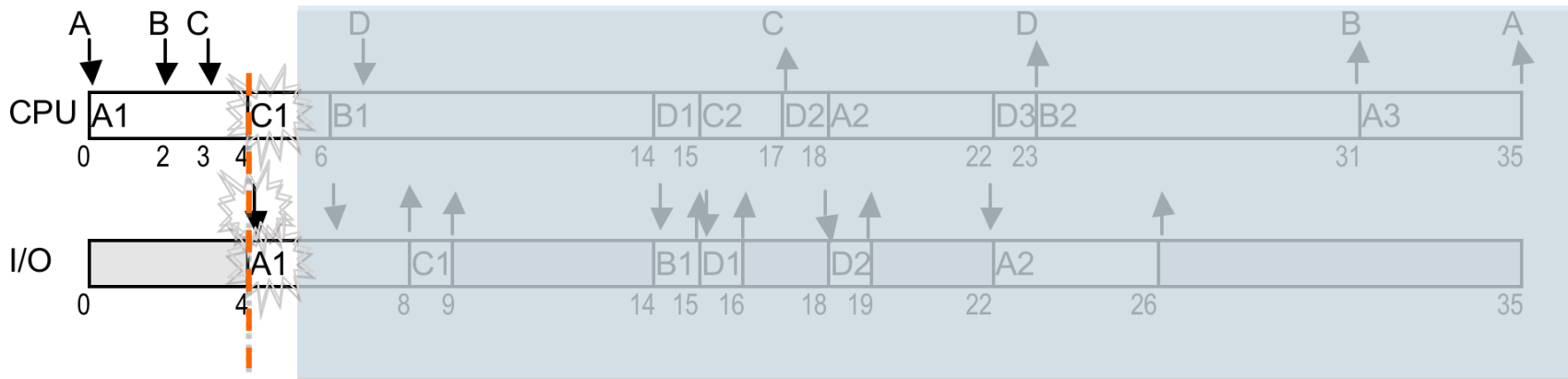
## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.2 Shortest-Process-First (SPF)

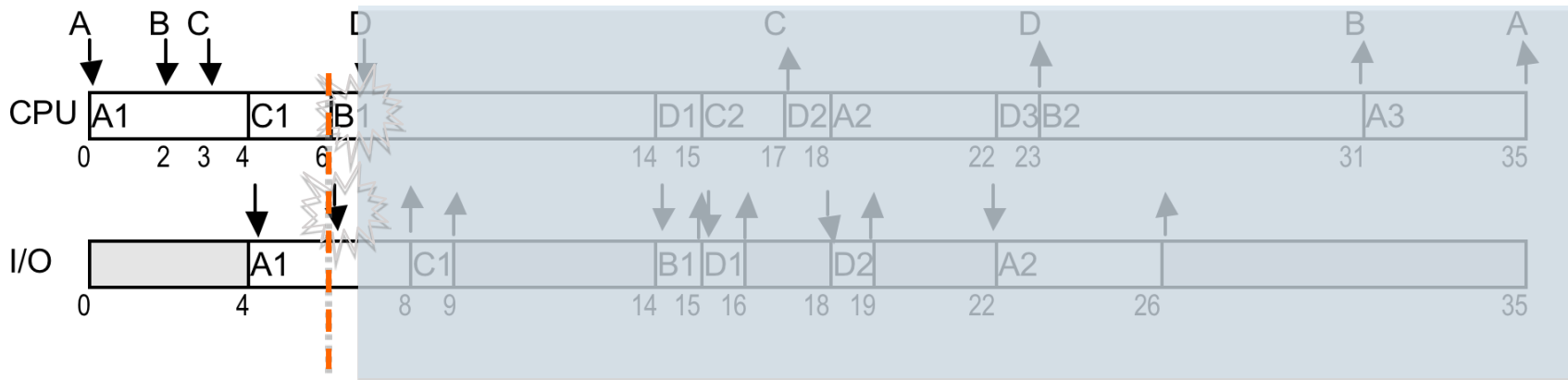
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

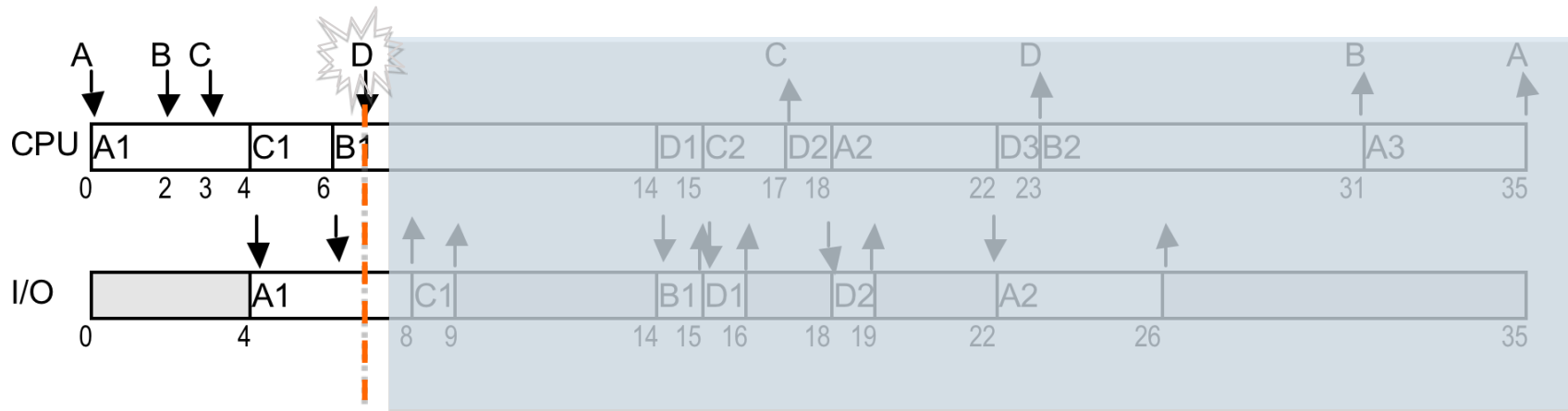
RQ: B:8



## 2.3.2 Shortest-Process-First (SPF)

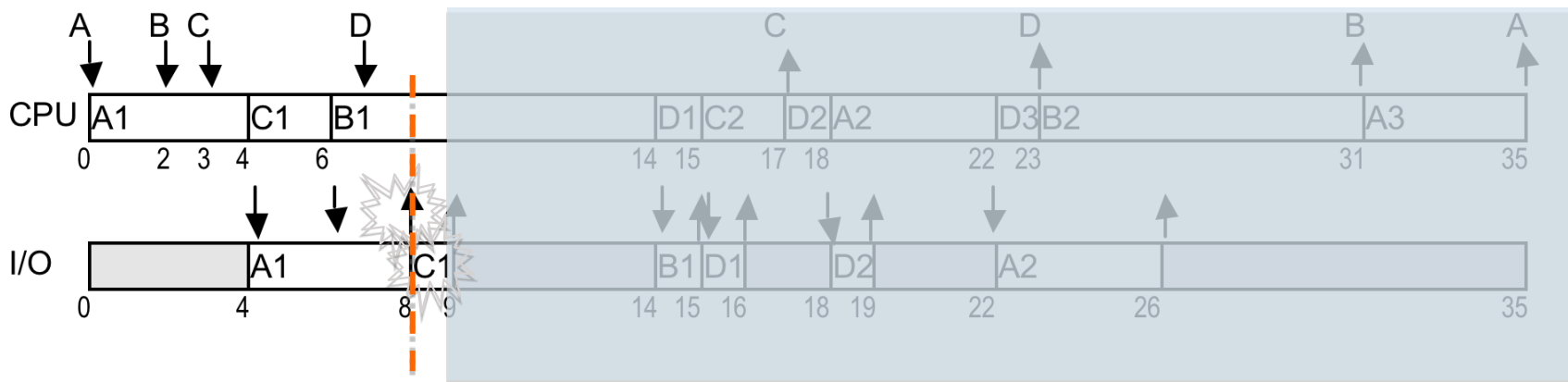
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

RQ:



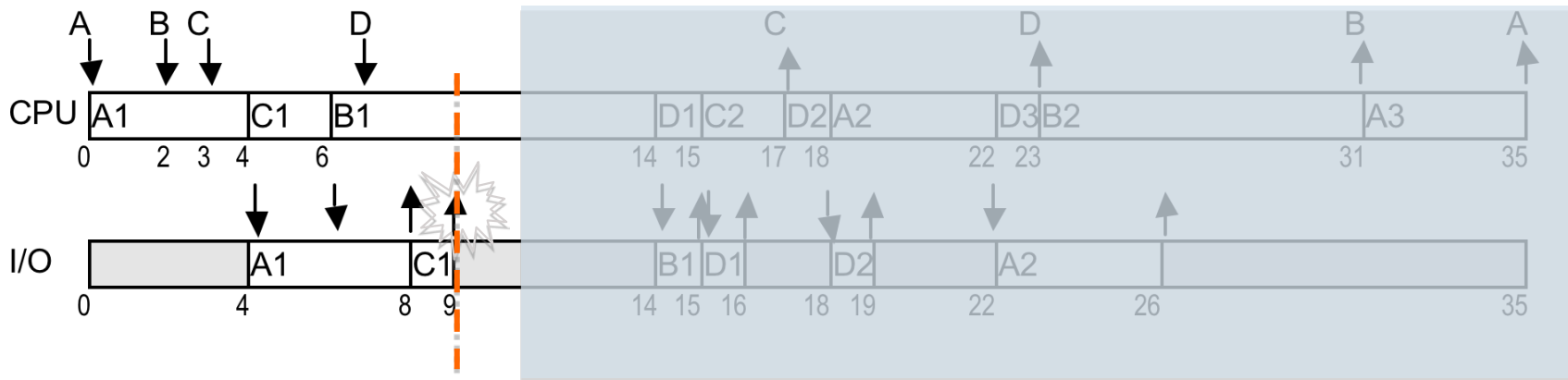
## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

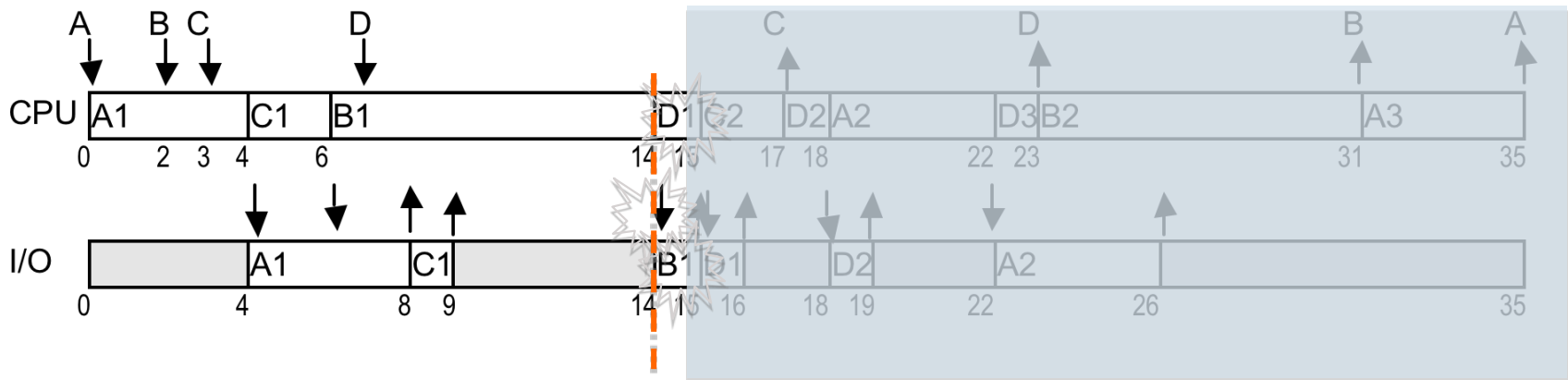




## 2.3.2 Shortest-Process-First (SPF)

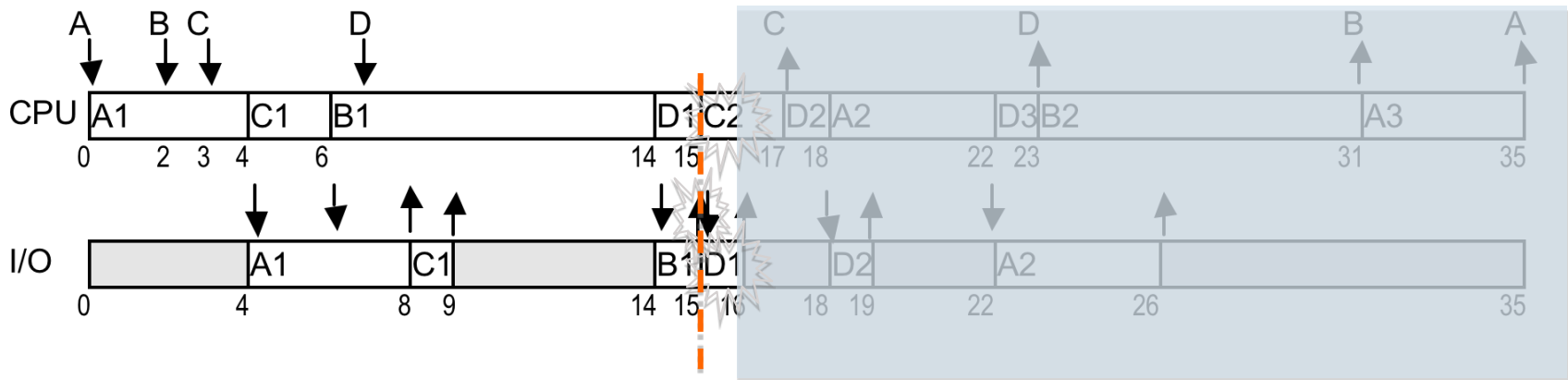
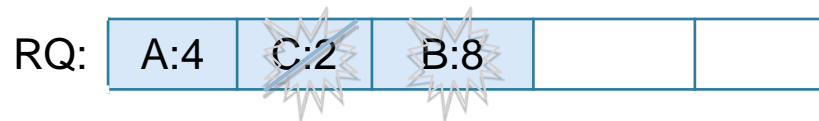
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

RQ: D:1 A:4 C:2



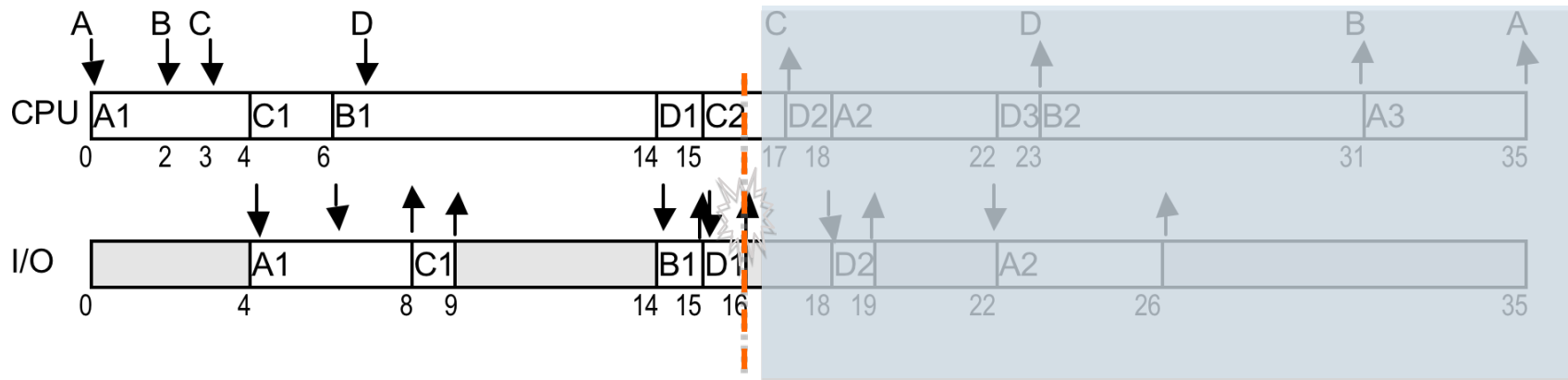
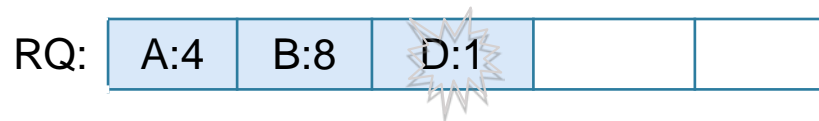
## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.2 Shortest-Process-First (SPF)

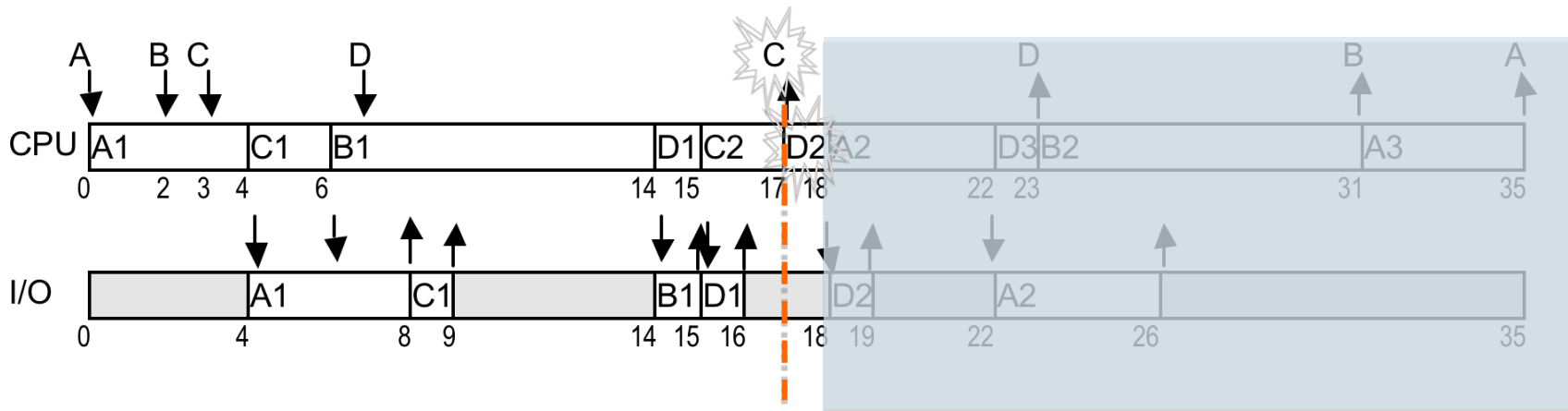
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

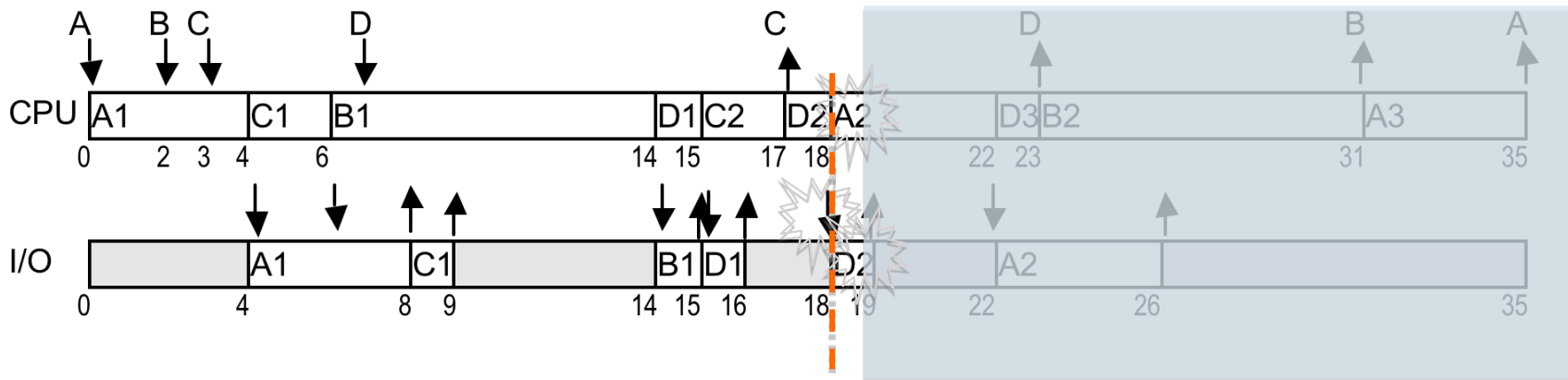
RQ: A:4 B:8 ~~D:1~~



## 2.3.2 Shortest-Process-First (SPF)

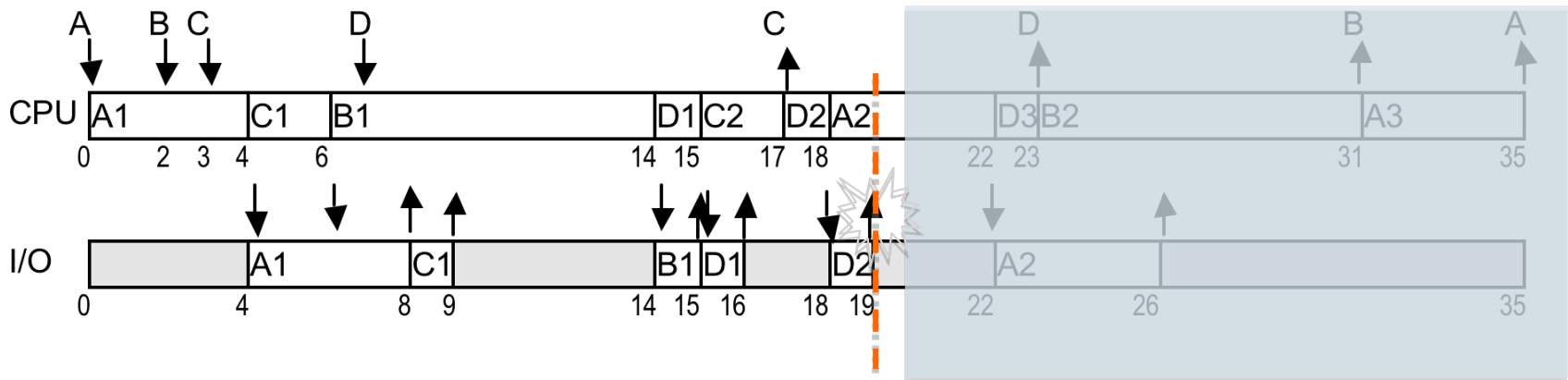
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

RQ: ~~A:4~~ B:8



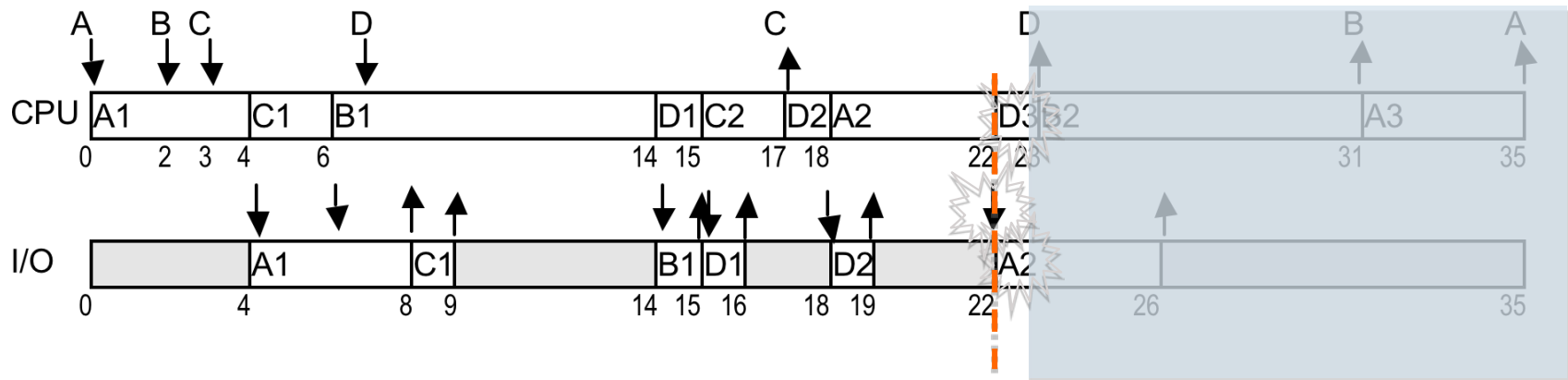
## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.2 Shortest-Process-First (SPF)

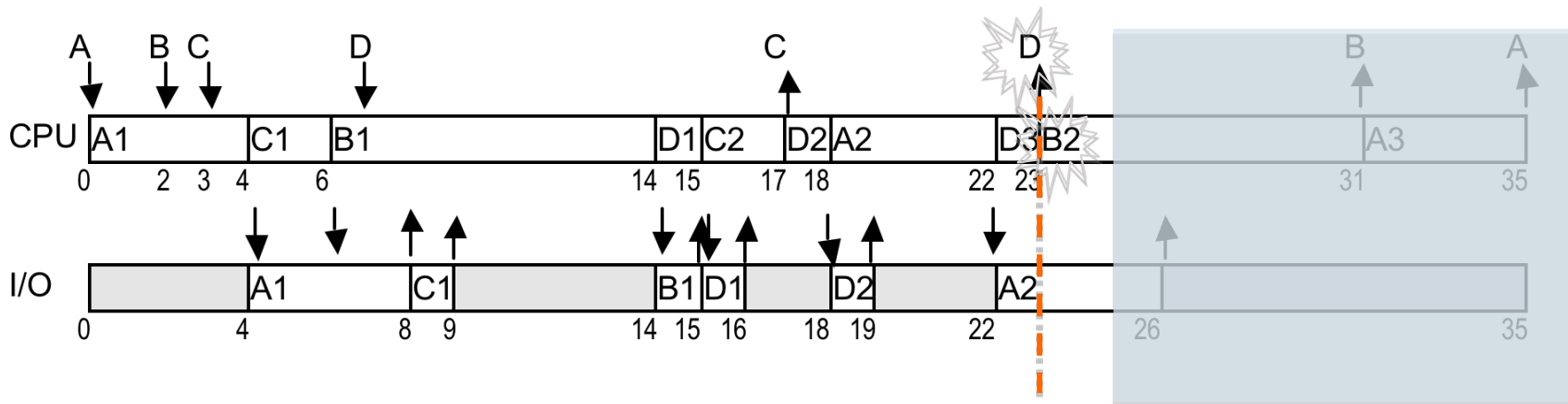
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1



## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

RQ: B:8

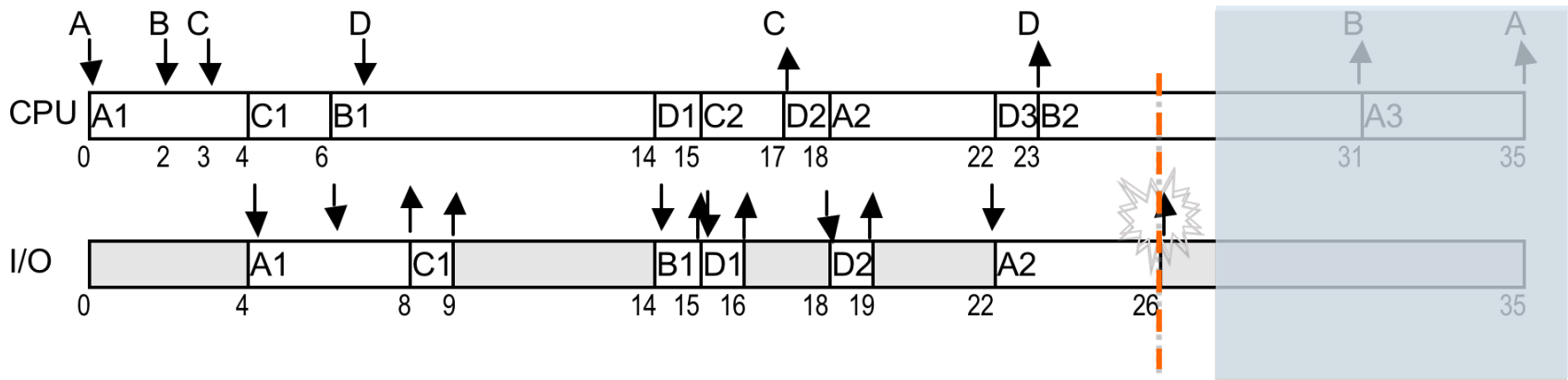




## 2.3.2 Shortest-Process-First (SPF)

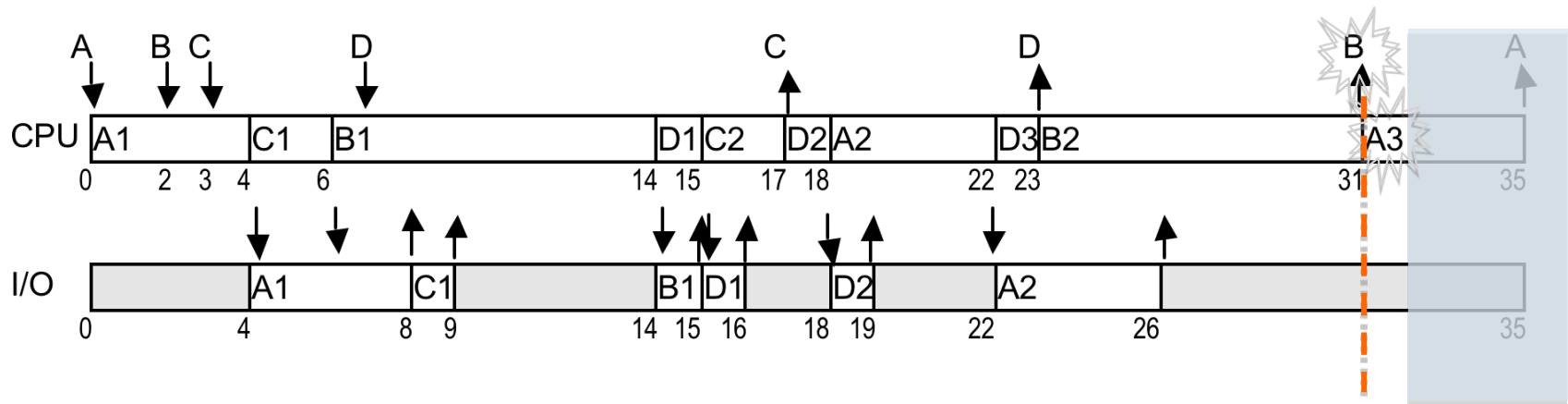
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

RQ: A:4



## 2.3.2 Shortest-Process-First (SPF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	..
C	3	2	1	2	-	-
D	7	1	1	1	1	1

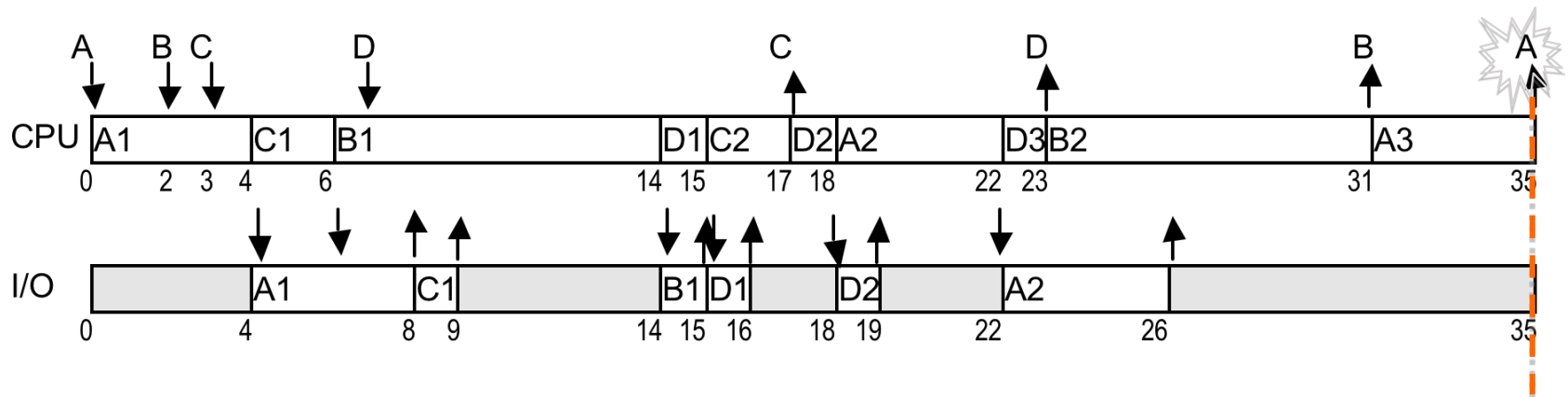


## 2.3.2 Shortest-Process-First (SPF)

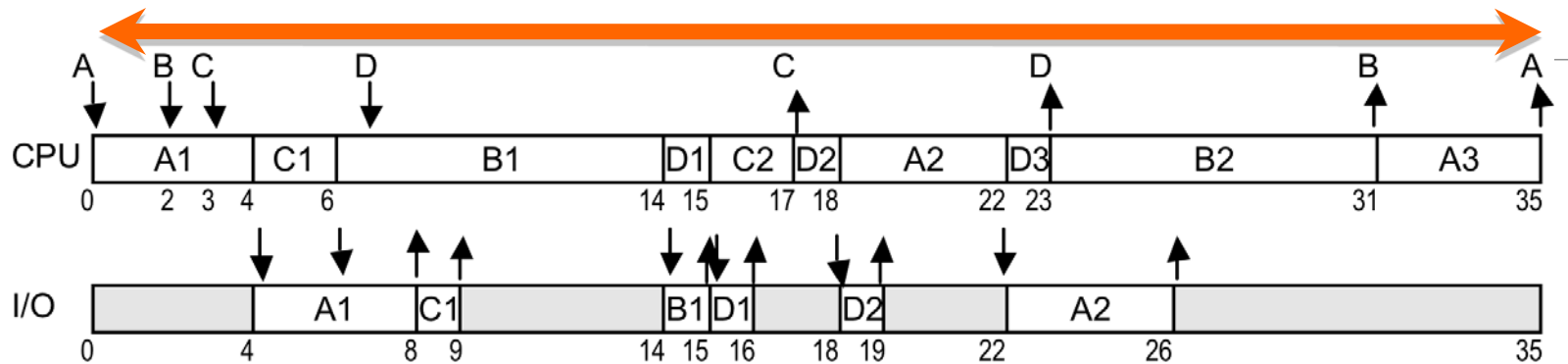
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4	4	4	4	4
B	2	8	1	8	-	..
C	3	2	1	2	-	-
D	7	1	1	1	1	1

RQ:

--	--	--	--	--

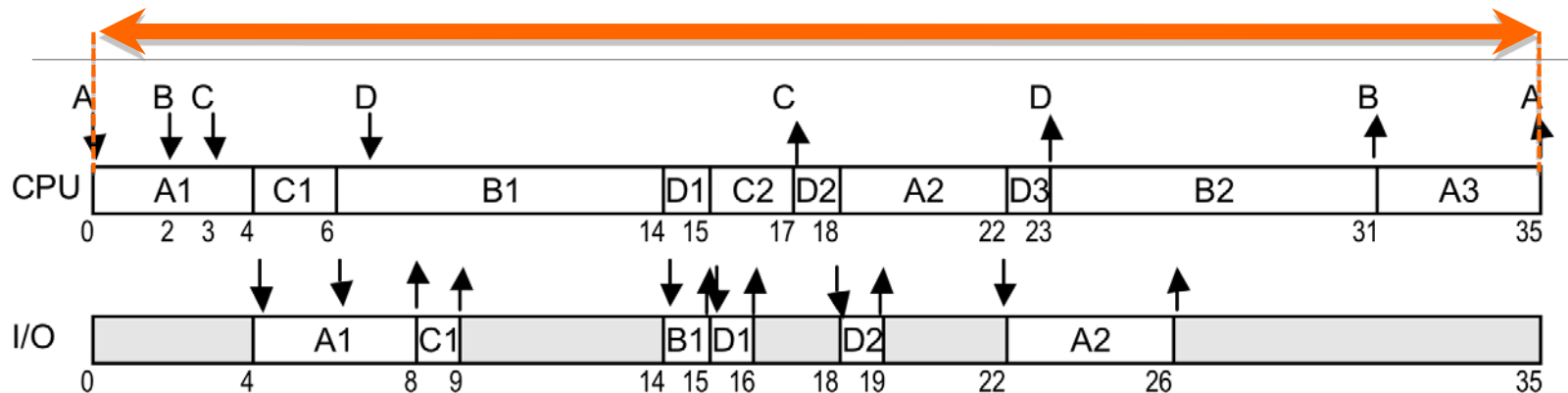


## 2.3.2 Shortest-Process-First (SPF)



- Processor utilization =  $(35 / 35) * 100 = 100 \%$
- Throughput =  $4 / 35 = 0.11$

## 2.3.2 Shortest-Process-First (SPF)



□ Turnaround time:

$$tat_A = 35 - 0 = 35$$

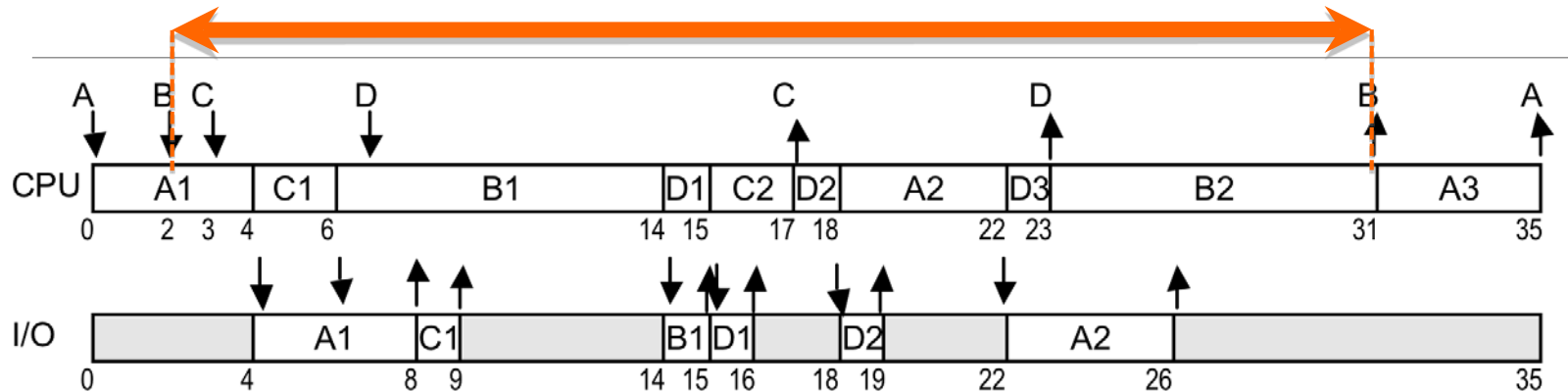
$$tat_B = 31 - 2 = 29$$

$$tat_C = 17 - 3 = 14$$

$$tat_D = 23 - 7 = 16$$

$$tat_{AVG} = (35 + 29 + 15 + 16) / 4 = 23.5$$

## 2.3.2 Shortest-Process-First (SPF)



□ Turnaround time:

$$tat_A = 35 - 0 = 35$$

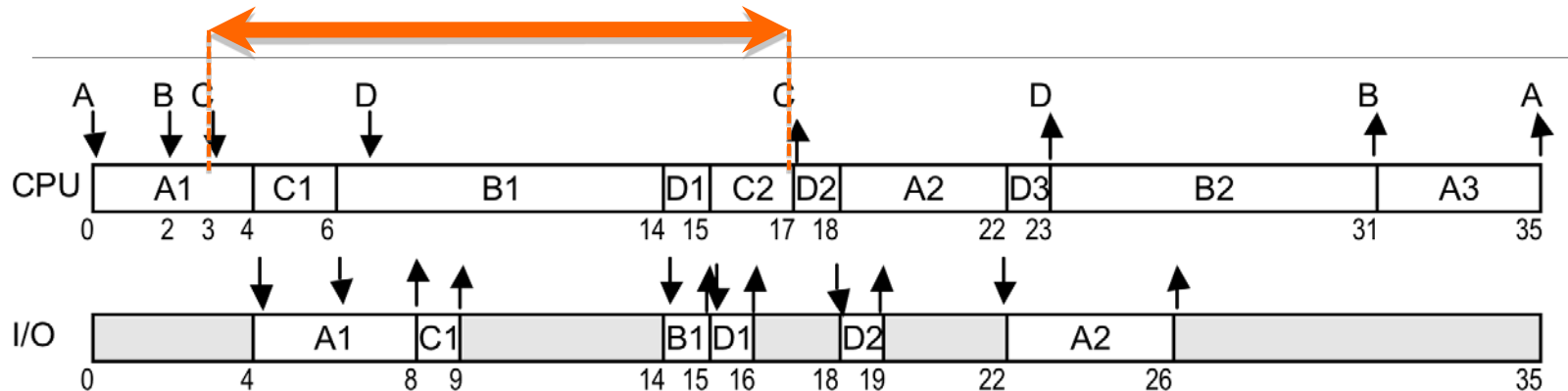
$$tat_B = 31 - 2 = 29$$

$$tat_C = 17 - 3 = 14$$

$$tat_D = 23 - 7 = 16$$

$$tat_{AVG} = (35 + 29 + 15 + 16) / 4 = 23.5$$

## 2.3.2 Shortest-Process-First (SPF)



□ Turnaround time:

$$\text{tat}_A = 35 - 0 = 35$$

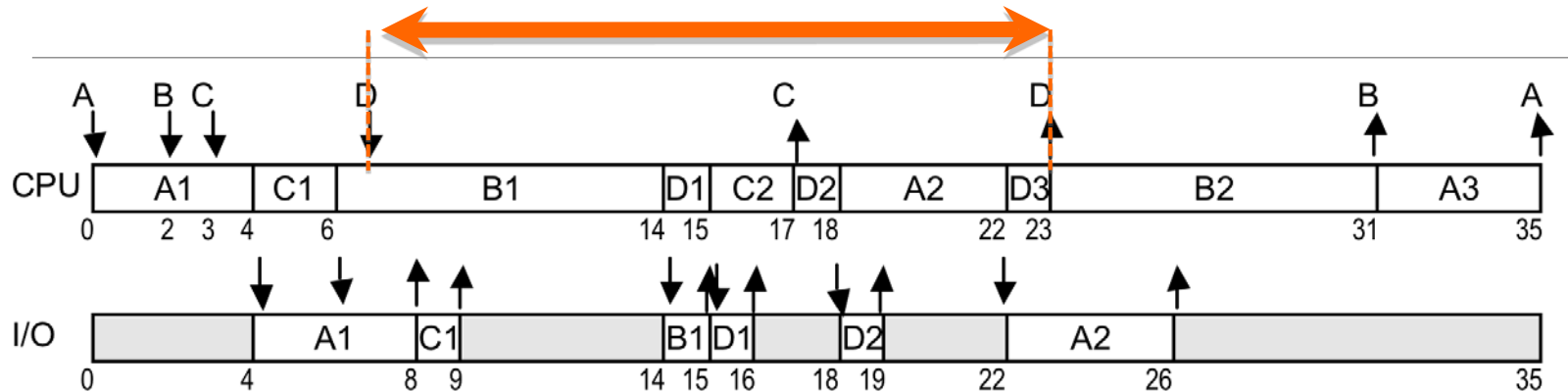
$$\text{tat}_B = 31 - 2 = 29$$

$$\text{tat}_C = 17 - 3 = 14$$

$$\text{tat}_D = 23 - 7 = 16$$

$$\text{tat}_{\text{AVG}} = (35 + 29 + 15 + 16) / 4 = 23.5$$

## 2.3.2 Shortest-Process-First (SPF)



□ Turnaround time:

$$tat_A = 35 - 0 = 35$$

$$tat_B = 31 - 2 = 29$$

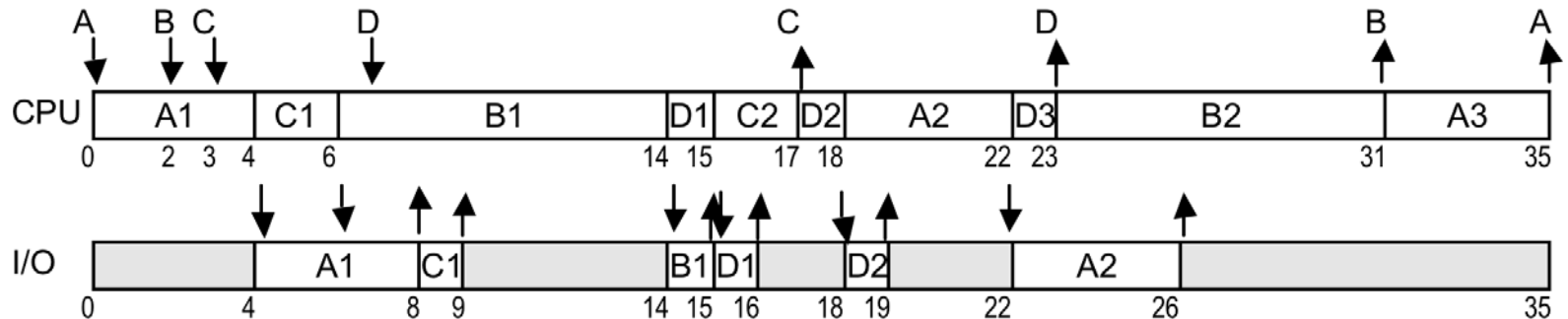
$$tat_C = 17 - 3 = 14$$

$$tat_D = 23 - 7 = 16$$

$$tat_{AVG} = (35 + 29 + 15 + 16) / 4 = 23.5$$



## 2.3.2 Shortest-Process-First (SPF)



□ Turnaround time:

$$tat_A = 35 - 0 = 35$$

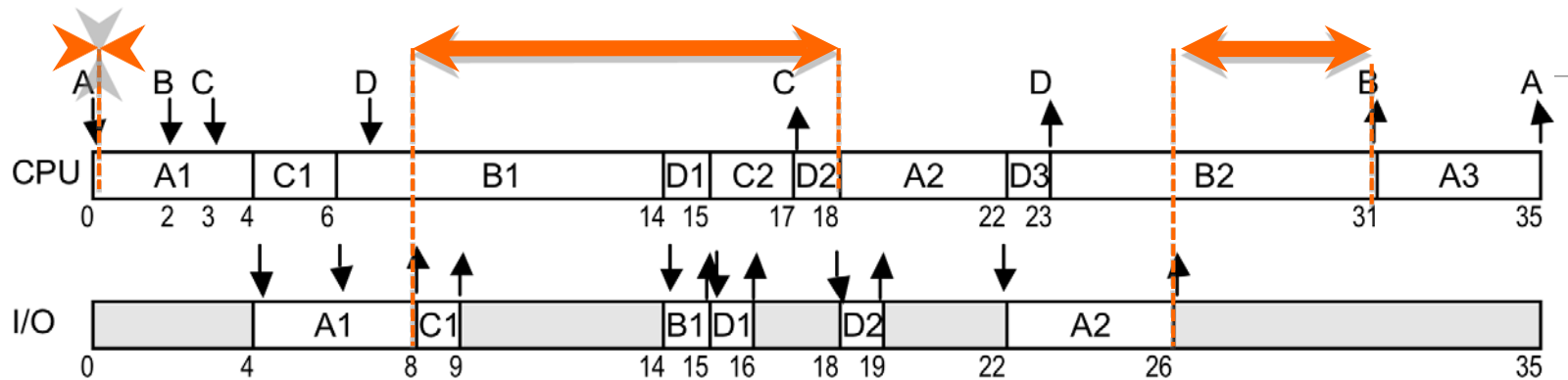
$$tat_B = 31 - 2 = 29$$

$$tat_C = 17 - 3 = 14$$

$$tat_D = 23 - 7 = 16$$

$$tat_{AVG} = (35 + 29 + 15 + 16) / 4 = 23.5$$

## 2.3.2 Shortest-Process-First (SPF)



□ Waiting time:

$$wt_A = (0 - 0) + (18 - 8) + (31 - 26) = 15$$

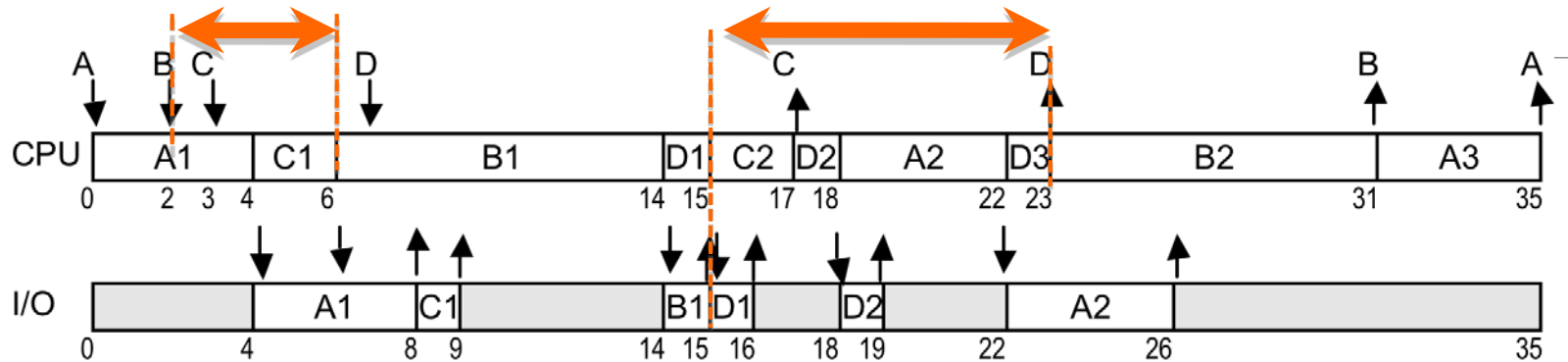
$$wt_B = (6 - 2) + (23 - 15) = 12$$

$$wt_C = (4 - 3) + (15 - 9) = 7$$

$$wt_D = (14 - 7) + (17 - 16) + (22 - 19) = 11$$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$

## 2.3.2 Shortest-Process-First (SPF)



### □ Waiting time:

$$wt_A = (0 - 0) + (18 - 8) + (31 - 26) = 15$$

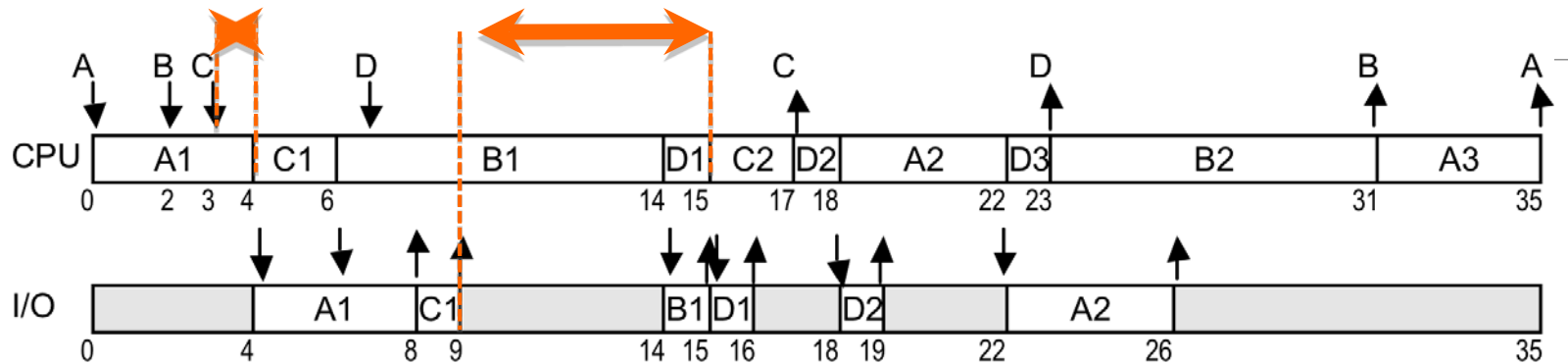
$$wt_B = (6 - 2) + (23 - 15) = 12$$

$$wt_C = (4 - 3) + (15 - 9) = 7$$

$$wt_D = (14 - 7) + (17 - 16) + (22 - 19) = 11$$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$

## 2.3.2 Shortest-Process-First (SPF)



□ Waiting time:

$$wt_A = (0 - 0) + (18 - 8) + (31 - 26) = 15$$

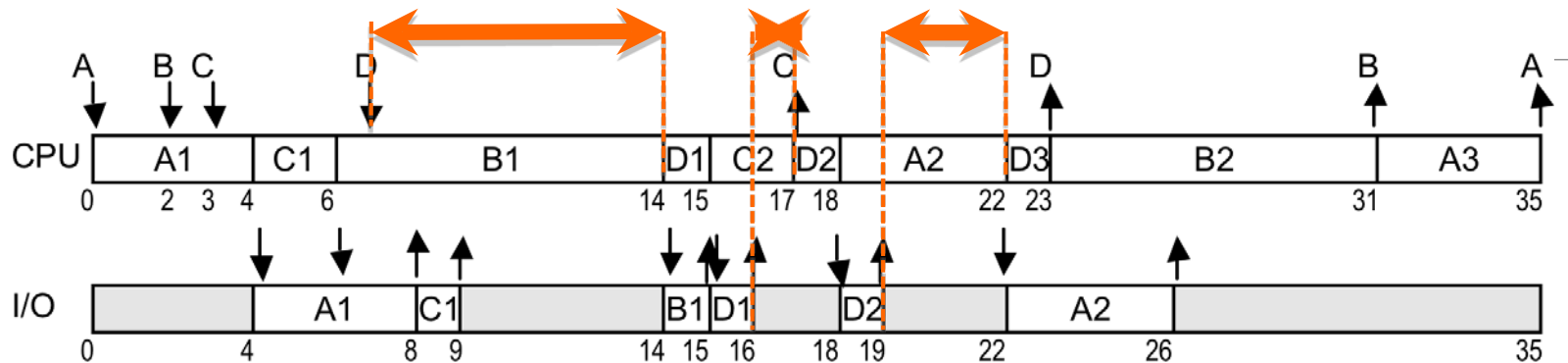
$$wt_B = (6 - 2) + (23 - 15) = 12$$

$$wt_C = (4 - 3) + (15 - 9) = 7$$

$$wt_D = (14 - 7) + (17 - 16) + (22 - 19) = 11$$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$

## 2.3.2 Shortest-Process-First (SPF)



### Waiting time:

$$wt_A = (0 - 0) + (18 - 8) + (31 - 26) = 15$$

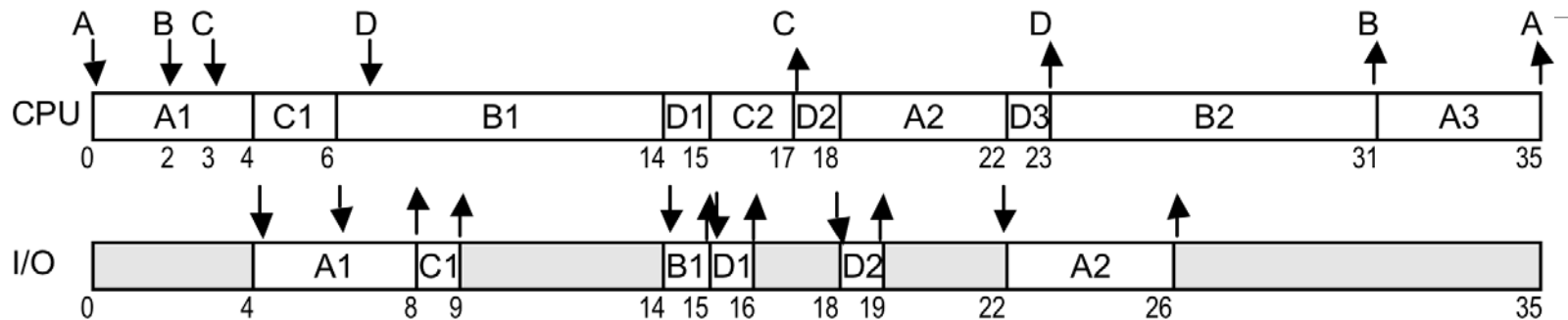
$$wt_B = (6 - 2) + (23 - 15) = 12$$

$$wt_C = (4 - 3) + (15 - 9) = 7$$

$$wt_D = (14 - 7) + (17 - 16) + (22 - 19) = 11$$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$

## 2.3.2 Shortest-Process-First (SPF)



### Waiting time:

$$wt_A = (0 - 0) + (18 - 8) + (31 - 26) = 15$$

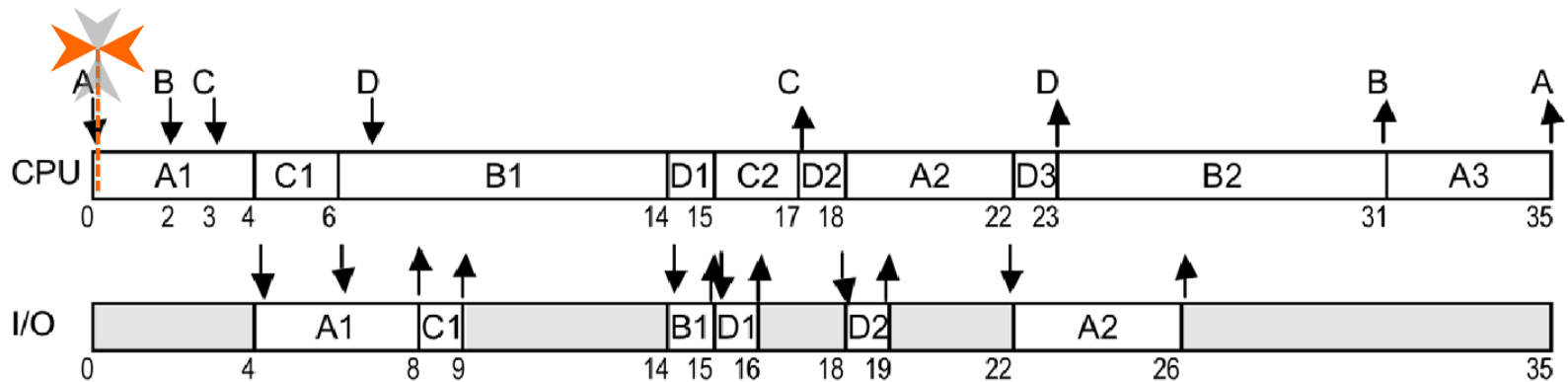
$$wt_B = (6 - 2) + (23 - 15) = 12$$

$$wt_C = (4 - 3) + (15 - 9) = 7$$

$$wt_D = (14 - 7) + (17 - 16) + (22 - 19) = 11$$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$

## 2.3.2 Shortest-Process-First (SPF)



□ Response time:

$$rt_A = 0 - 0 = 0$$

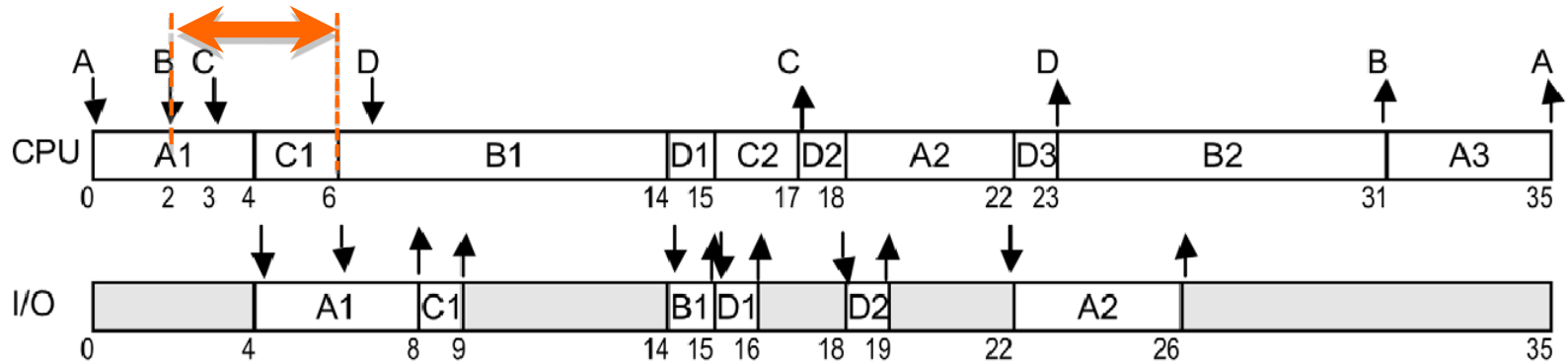
$$rt_B = 6 - 2 = 4$$

$$rt_C = 4 - 3 = 1$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 4 + 1 + 7) / 4 = 3$$

## 2.3.2 Shortest-Process-First (SPF)



□ Response time:

$$rt_A = 0 - 0 = 0$$

$$rt_B = 6 - 2 = 4$$

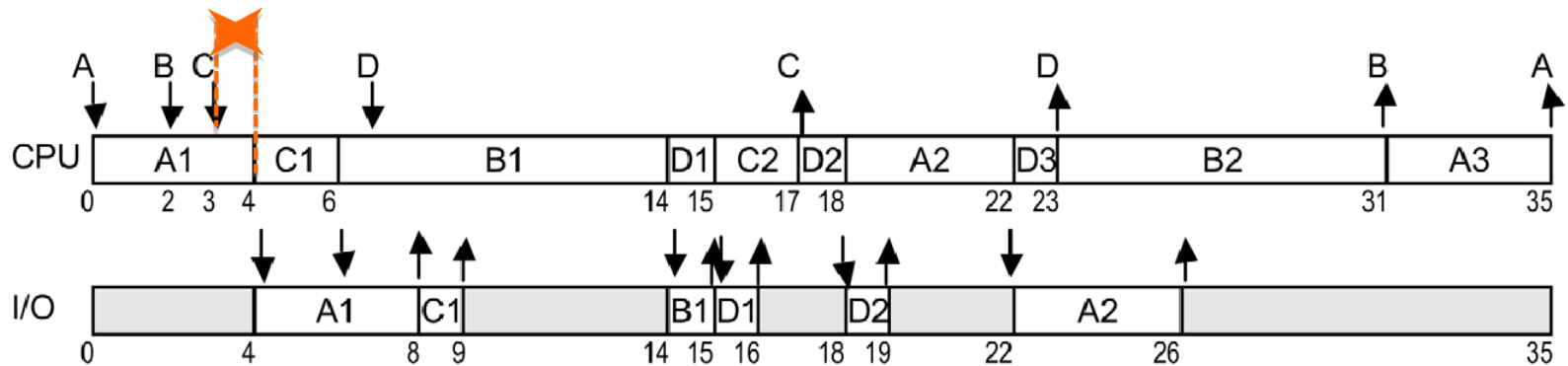
$$rt_C = 4 - 3 = 1$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 4 + 1 + 7) / 4 = 3$$



## 2.3.2 Shortest-Process-First (SPF)



□ Response time:

$$rt_A = 0 - 0 = 0$$

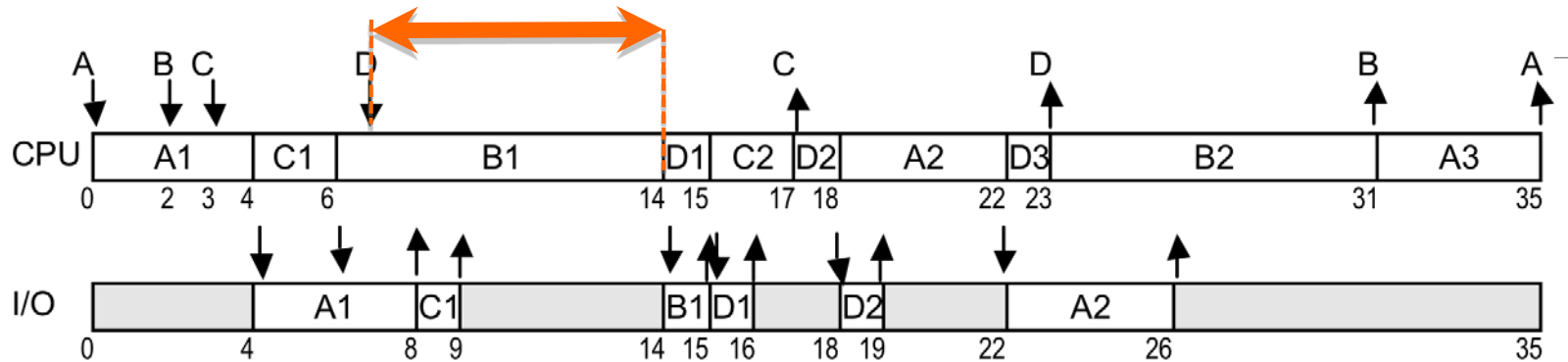
$$rt_B = 6 - 2 = 4$$

$$rt_C = 4 - 3 = 1$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 4 + 1 + 7) / 4 = 3$$

## 2.3.2 Shortest-Process-First (SPF)



□ Response time:

$$rt_A = 0 - 0 = 0$$

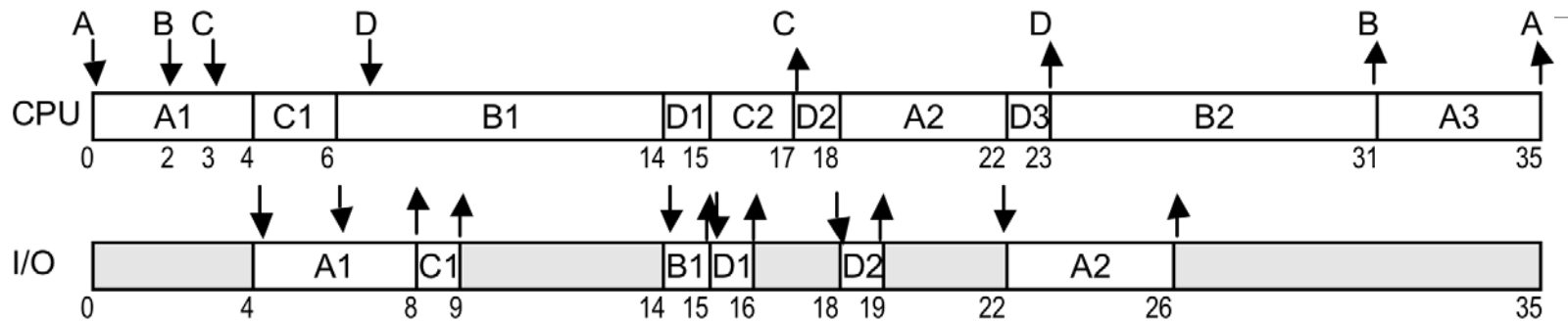
$$rt_B = 6 - 2 = 4$$

$$rt_C = 4 - 3 = 1$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 4 + 1 + 7) / 4 = 3$$

## 2.3.2 Shortest-Process-First (SPF)



□ Response time:

$$rt_A = 0 - 0 = 0$$

$$rt_B = 6 - 2 = 4$$

$$rt_C = 4 - 3 = 1$$

$$rt_D = 14 - 7 = 7$$

$$rt_{AVG} = (0 + 4 + 1 + 7) / 4 = 3$$



# ORDONNANCEMENT AVEC RÉQUISITION

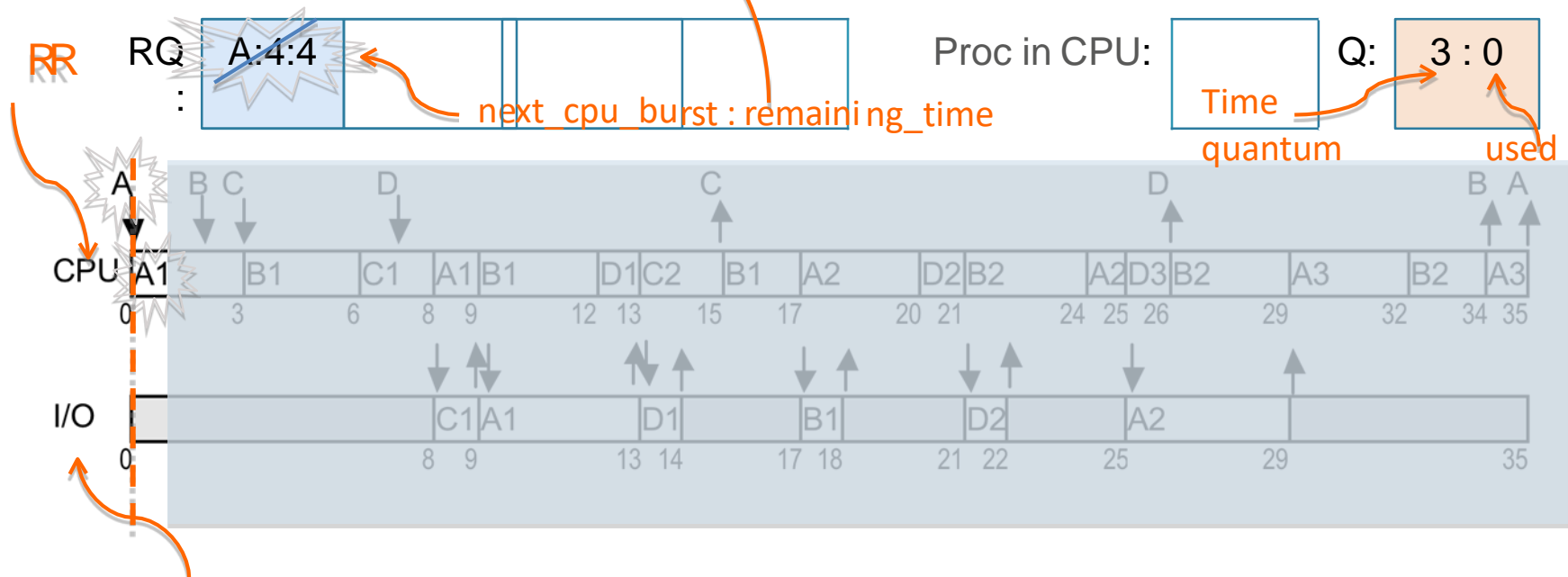
## Ordonnancement avec réquisition (préemptifs)

- Pour s'assurer qu'aucun processus ne s'exécute pendant trop de temps, une horloge génère périodiquement une interruption.
- A chaque interruption d'horloge, le système d'exploitation reprend la main et décide si le processus courant doit :
  - poursuivre son exécution ou
  - être suspendu pour laisser place à un autre.
- S'il décide de suspendre l'exécution au profit d'un autre, il doit d'abord sauvegarder l'état des registres du processeur avant de charger dans le registres les données du processus à lancer (commutation de contexte).
- Cette sauvegarde est nécessaire pour pouvoir poursuivre ultérieurement l'exécution du processus suspendu.

# ALGORITHME DU TOURNIQUET (ROUND ROBIN)

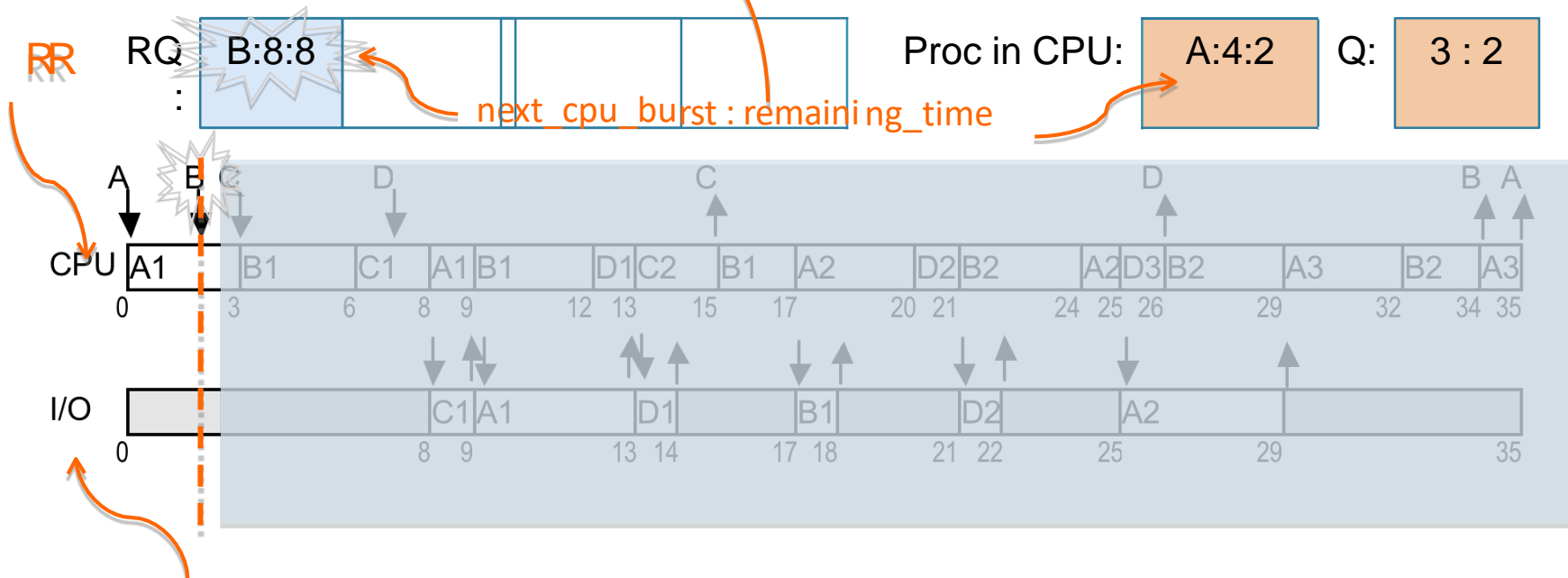
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>ère</sup> exec	1 <sup>s</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 4	4	4 : 4	4	4 : 4
B	2	8 : 8	1	8 : 8	-	-
C	3	2 : 2	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



# Algorithme du Tourniquet (Round Robin)

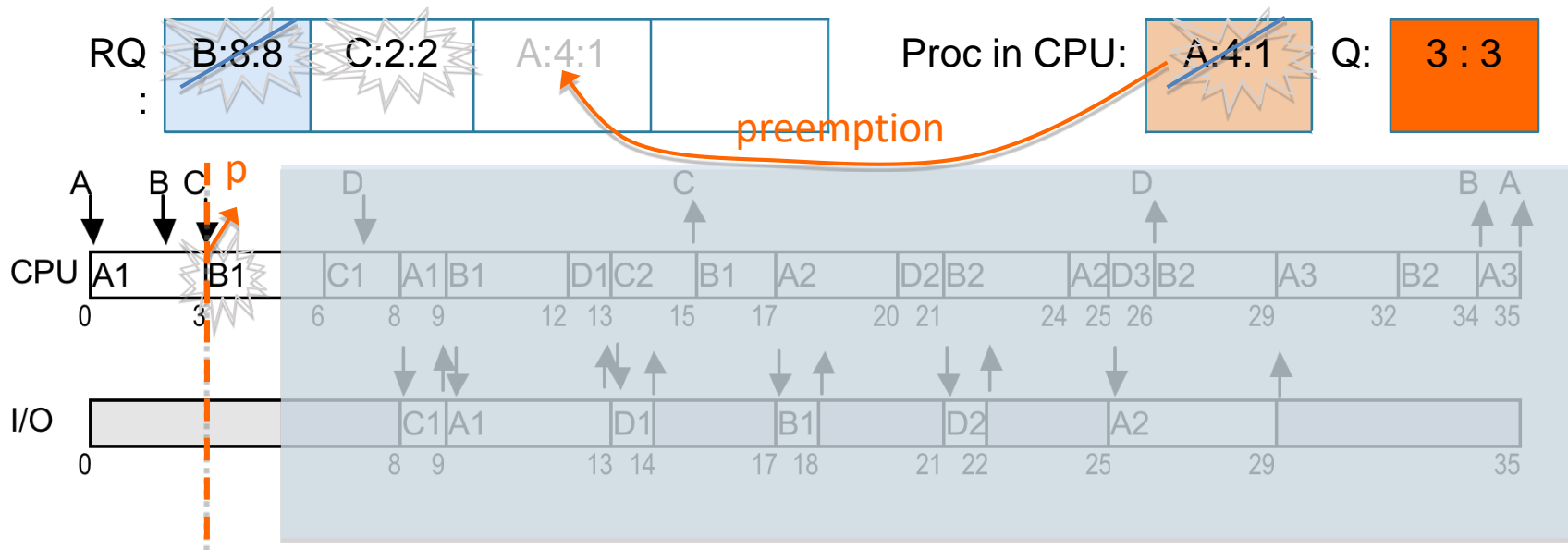
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 2	4	4 : 4	4	4 : 4
B	2	8 : 8	1	8 : 8	-	-
C	3	2 : 2	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1





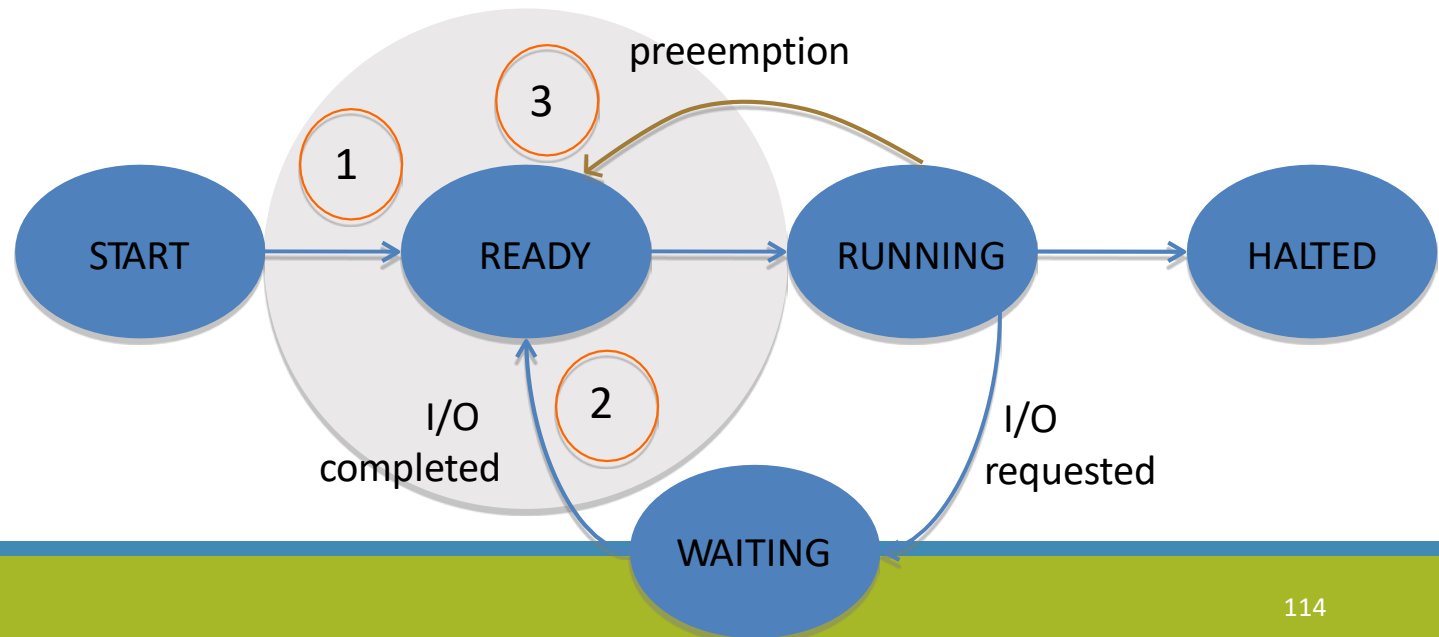
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 1	4	4 : 4	4	4 : 4
B	2	8 : 8	1	8 : 8	-	-
C	3	2 : 2	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



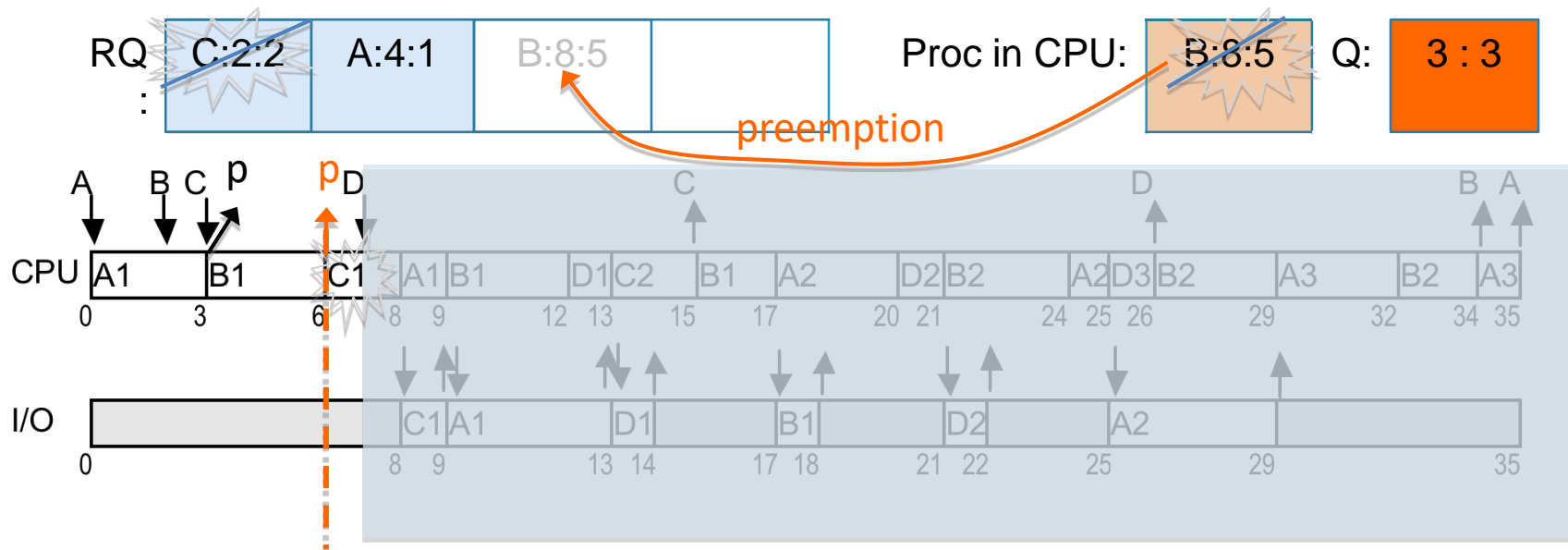
# Etats des processus

- Si plus d'un processus essayent d'entrer en même temps à la file d'attente des processus prêts, la priorité est donnée comme suit:
- 1: processus qui viennent d'être soumis
- 2: processus qui ont fini une opération d'E/s
- 3: Les processus qui ont été préemptés par le CPU



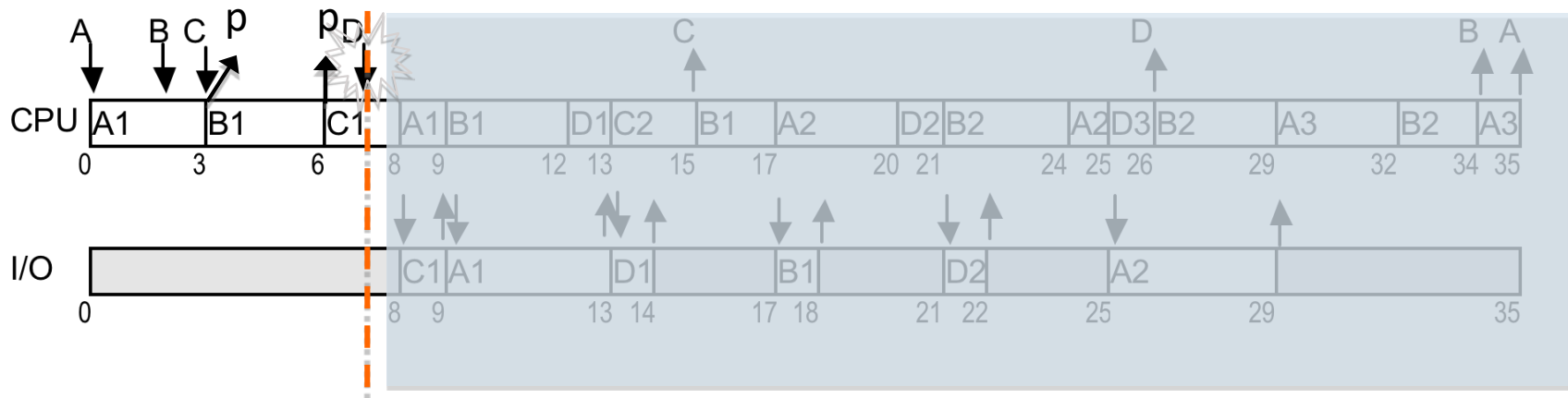
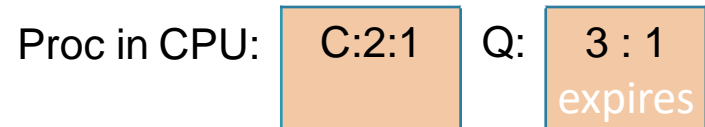
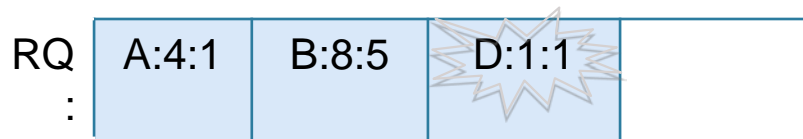
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 1	4	4 : 4	4	4 : 4
B	2	8 : 5	1	8 : 8	-	-
C	3	2 : 2	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



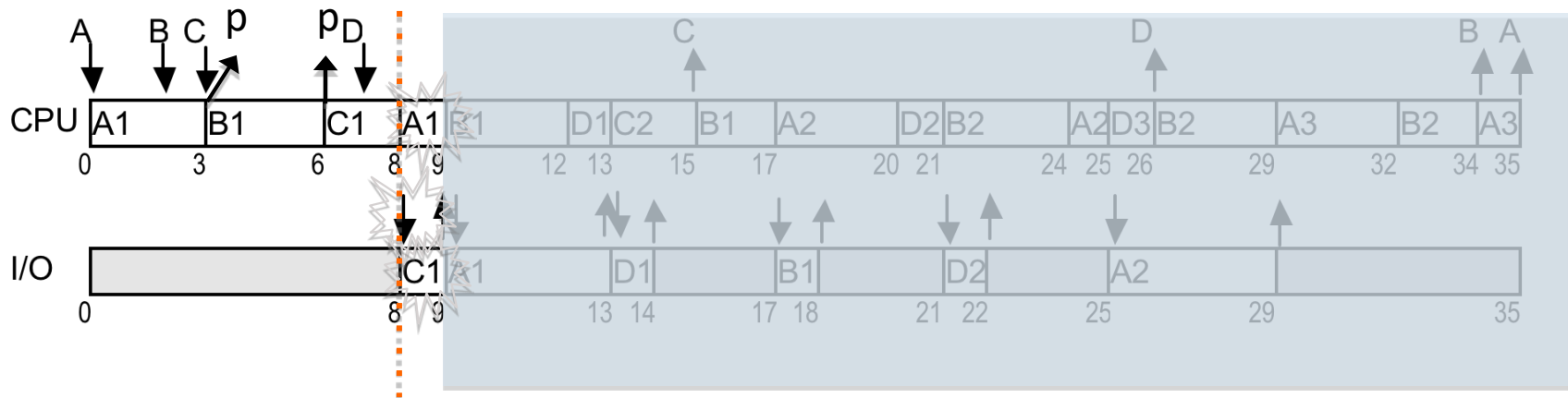
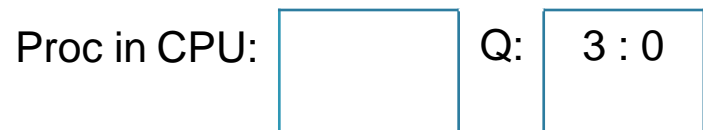
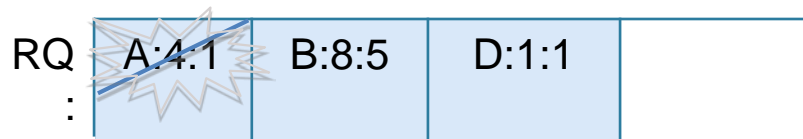
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 1	4	4 : 4	4	4 : 4
B	2	8 : 5	1	8 : 8	-	-
C	3	2 : 1	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



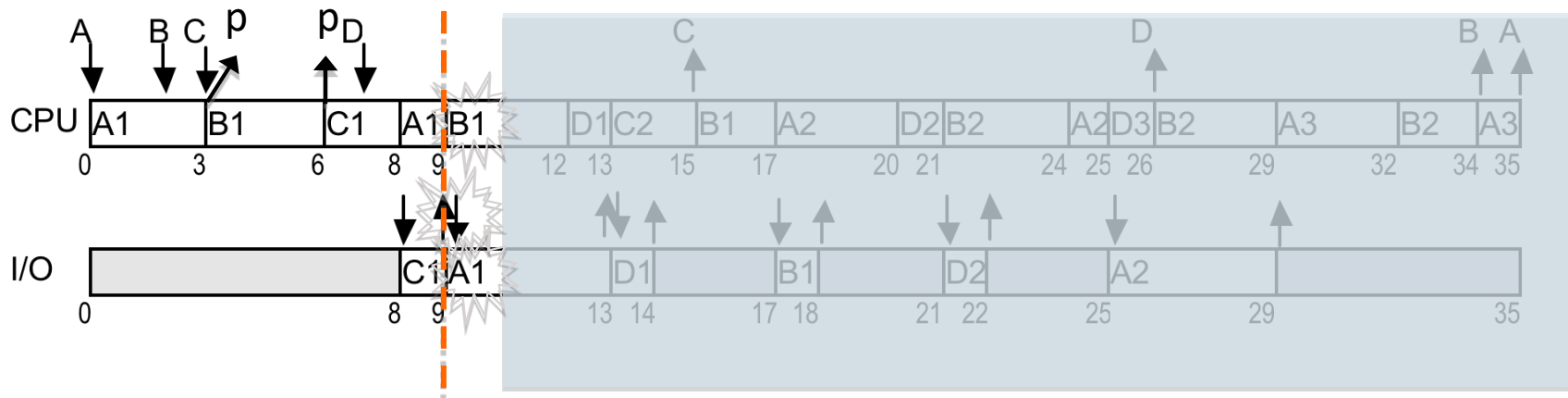
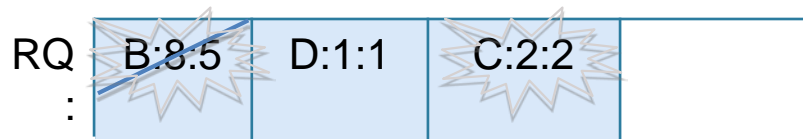
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 1	4	4 : 4	4	4 : 4
B	2	8 : 5	1	8 : 8	-	-
C	3	2 : 0	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



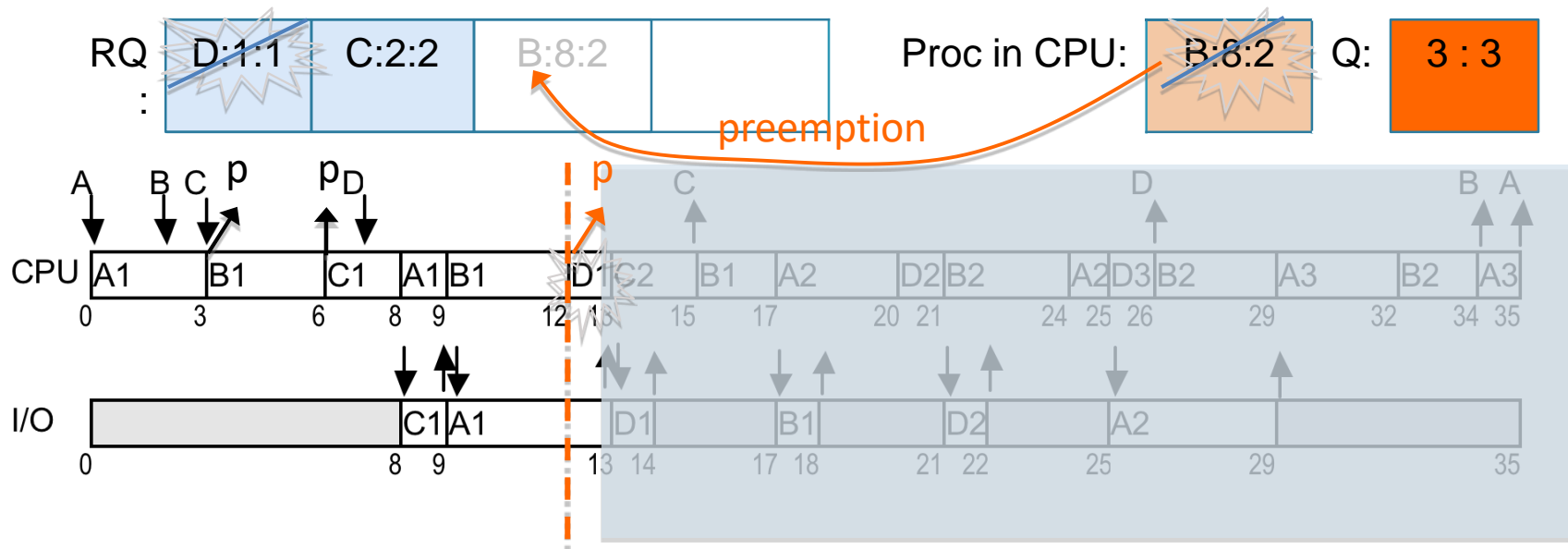
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 5	1	8 : 8	-	-
C	3	2 : 0	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 2	1	8 : 8	-	-
C	3	2 : 0	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 2	1	8 : 8	-	-
C	3	2 : 0	1	2 : 2	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1

RQ :

C:2:2

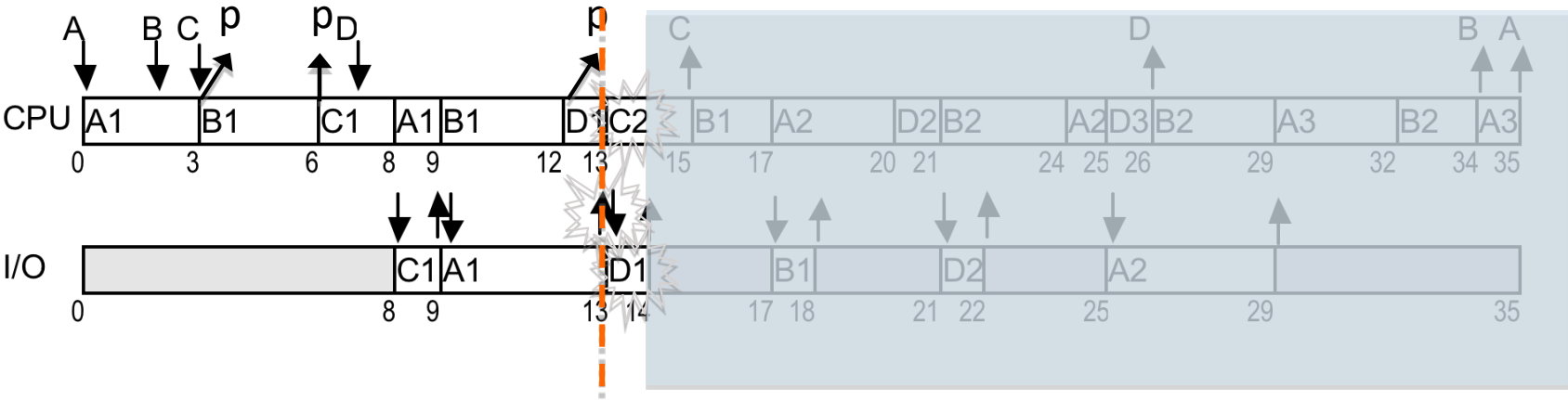
B:8:2

A:4:4

Proc in CPU:

Q:

3 : 0





# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 2	1	8 : 8	-	-
C	3	2 : 0	1	2 : 1	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1

RQ
:

B:8:2

A:4:4

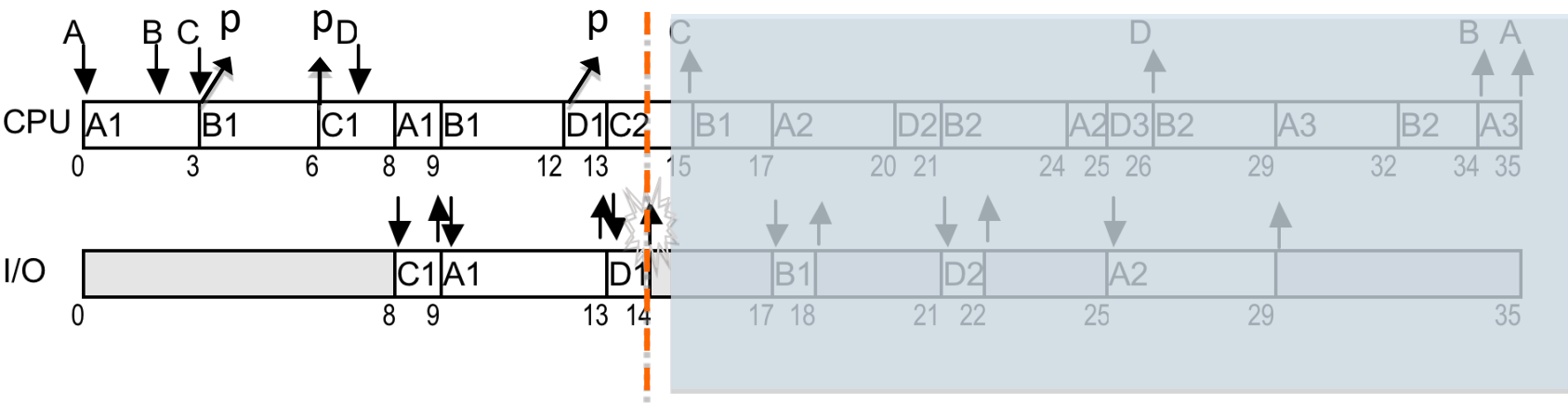
D:1:1

Proc in CPU:

C:2:1

Q:

3 : 1 expires



# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 2	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1

RQ :

B:8:2

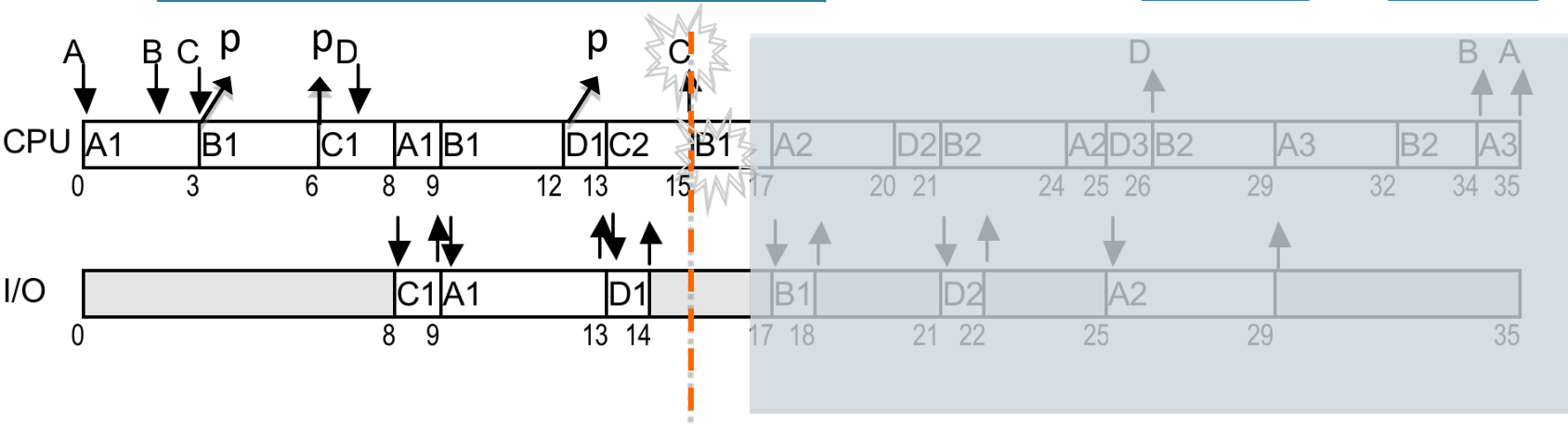
A:4:4

D:1:1

Proc in CPU:

Q:

3 : 0



# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 0	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1

RQ :

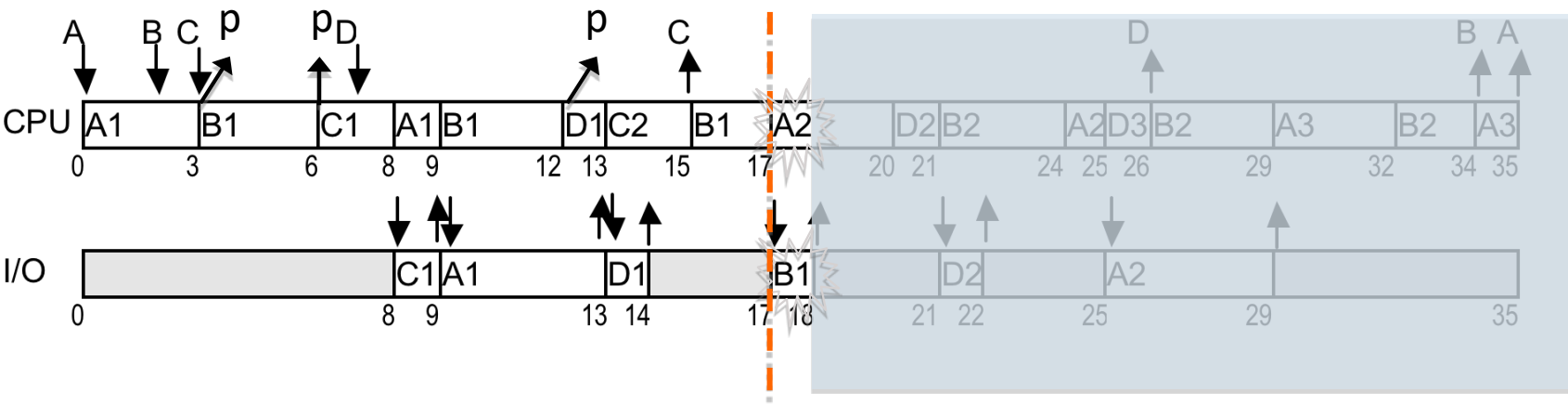
A:4:4

D:1:1

Proc in CPU:

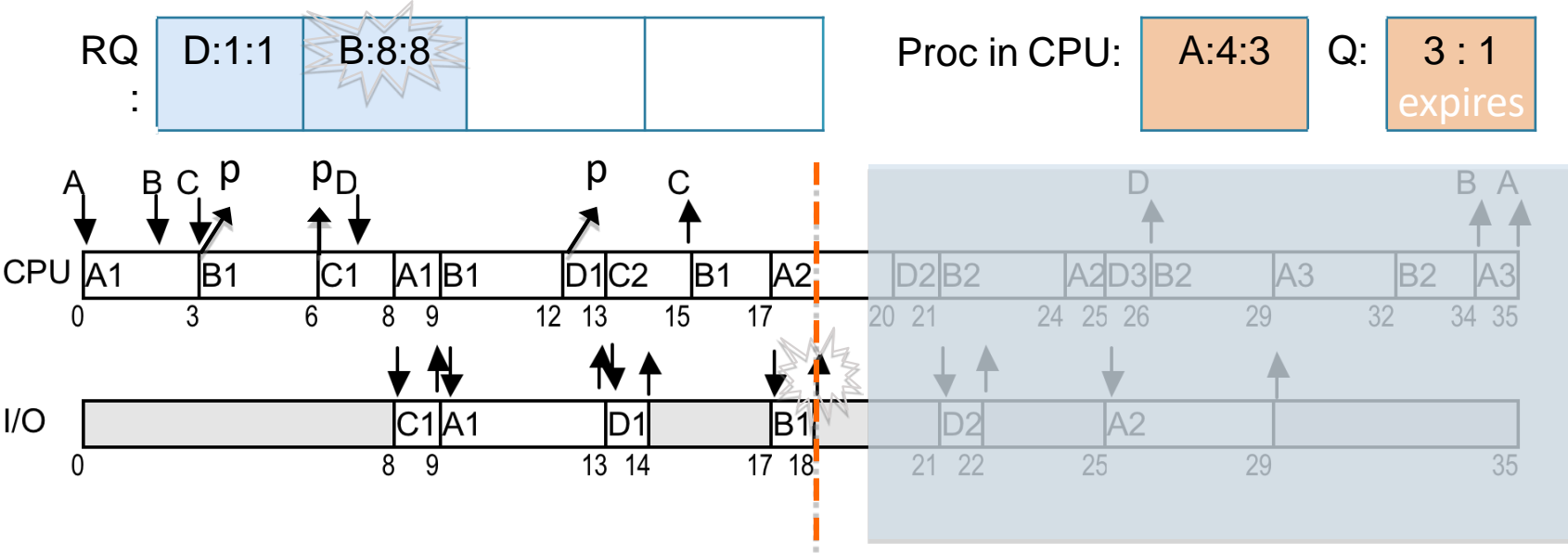
Q:

3 : 0



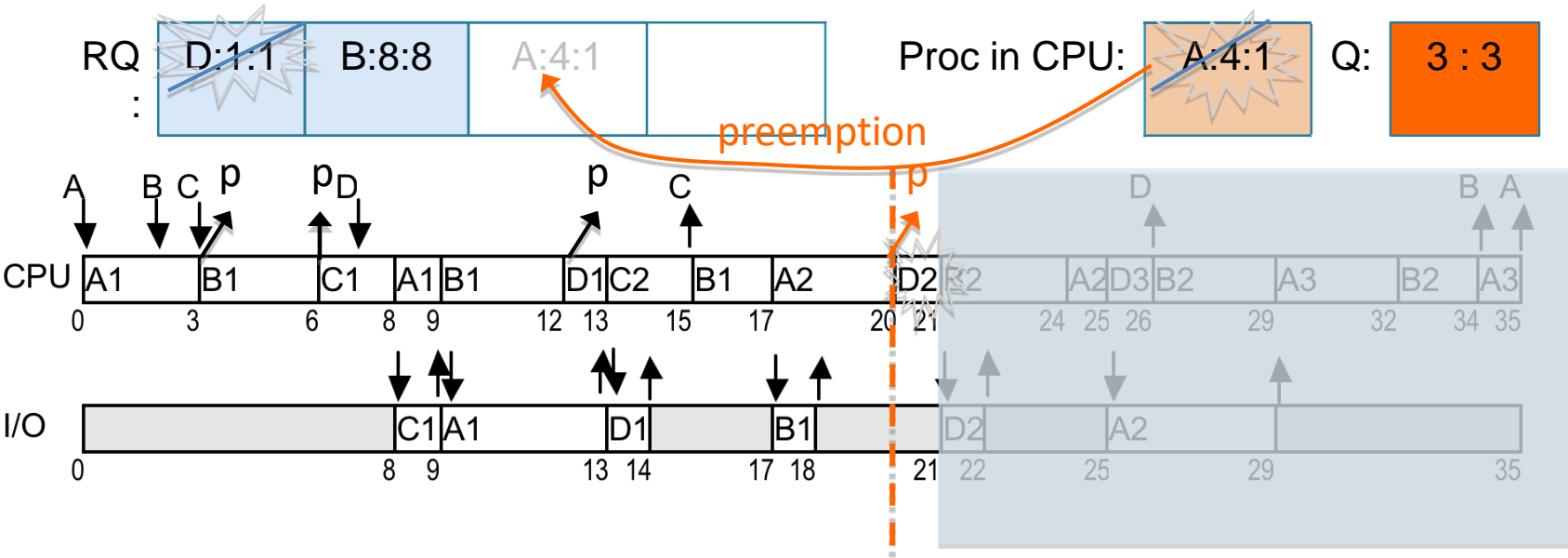
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 3	4	4 : 4
B	2	8 : 0	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1



# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 1	4	4 : 4
B	2	8 : 0	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1



# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 1	4	4 : 4
B	2	8 : 0	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 1

RQ :

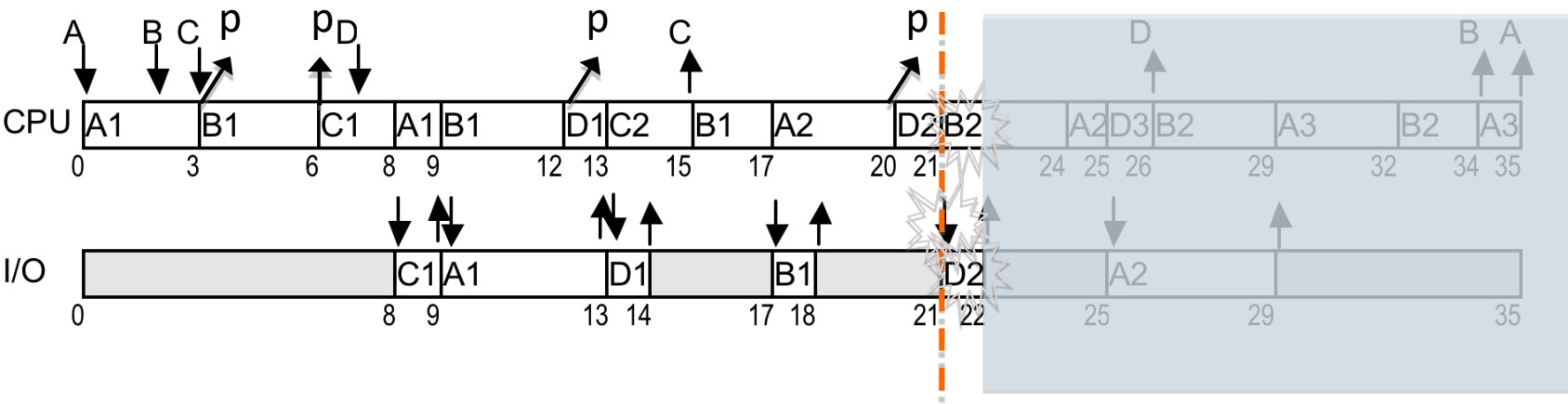
B:8.8

A:4:1

Proc in CPU:

Q:

3 : 0



# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 1	4	4 : 4
B	2	8 : 0	1	8 : 7	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 1

RQ :

A:4:1

D:1:1

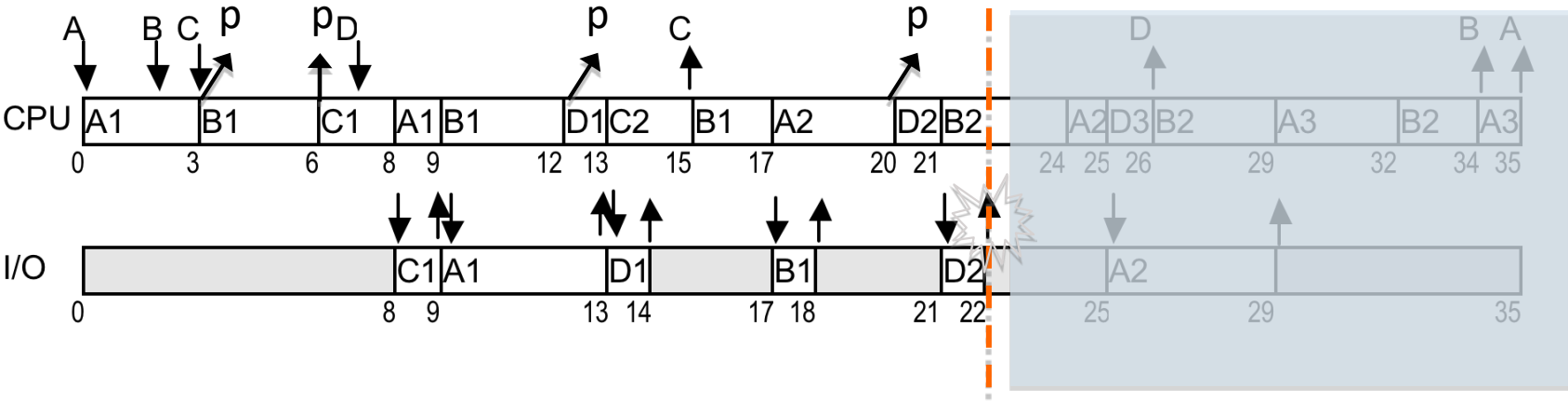
Proc in CPU:

B:8:7

Q:

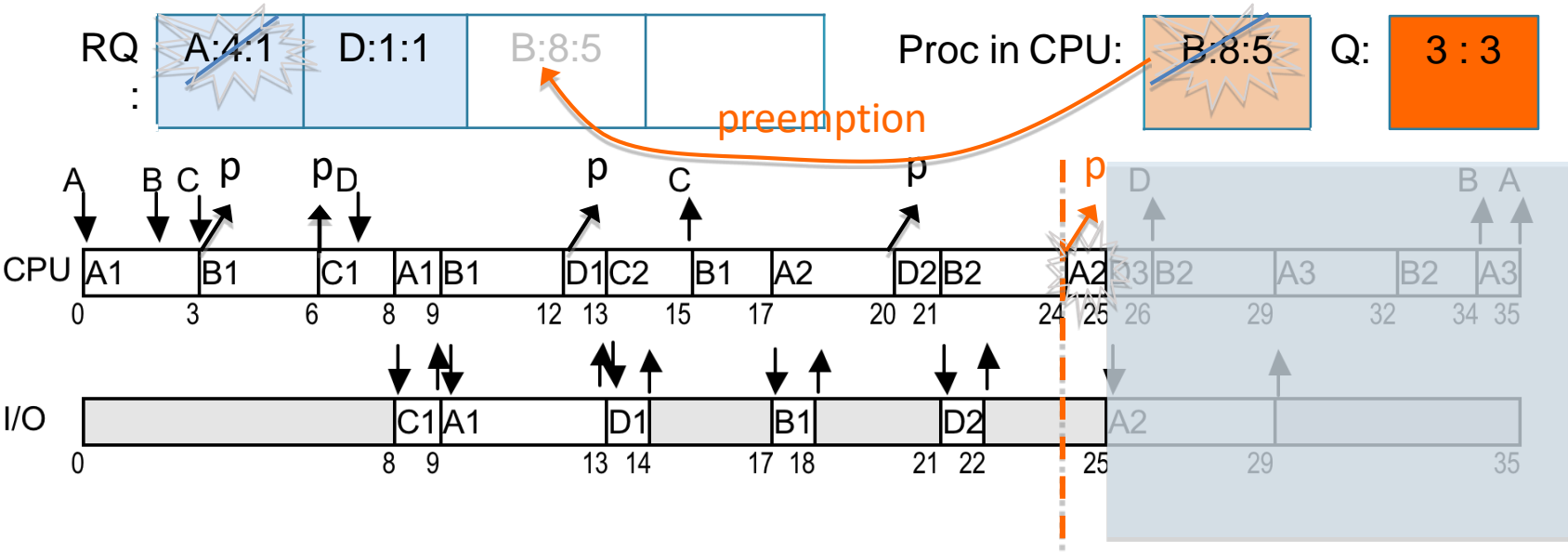
3 : 1

expires



# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 1	4	4 : 4
B	2	8 : 0	1	8 : 5	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 1





# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 4
B	2	8 : 0	1	8 : 5	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 1

RQ :

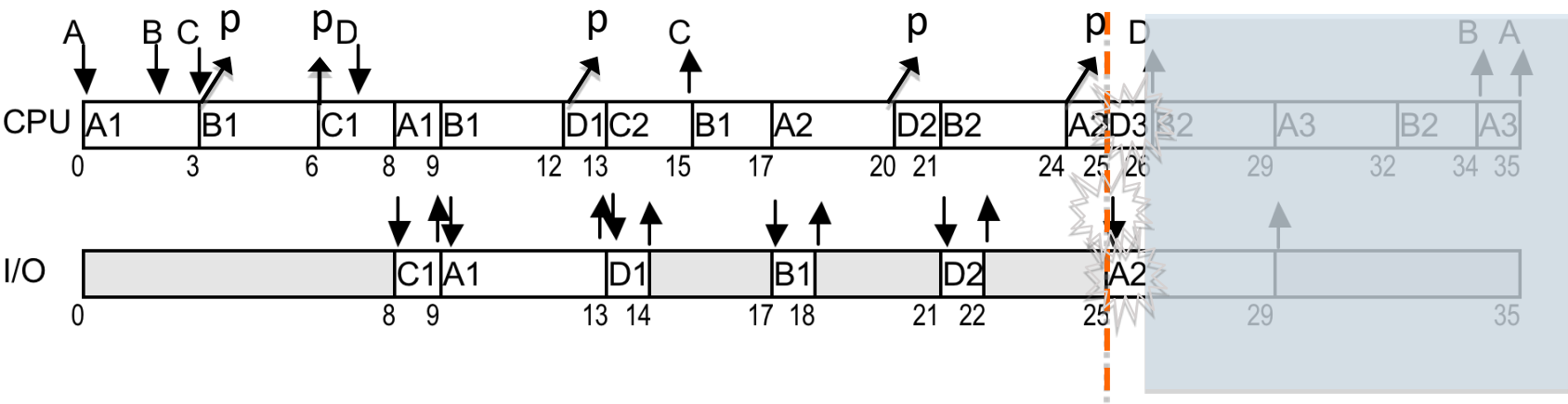
D:1:1

B:8:5

Proc in CPU:

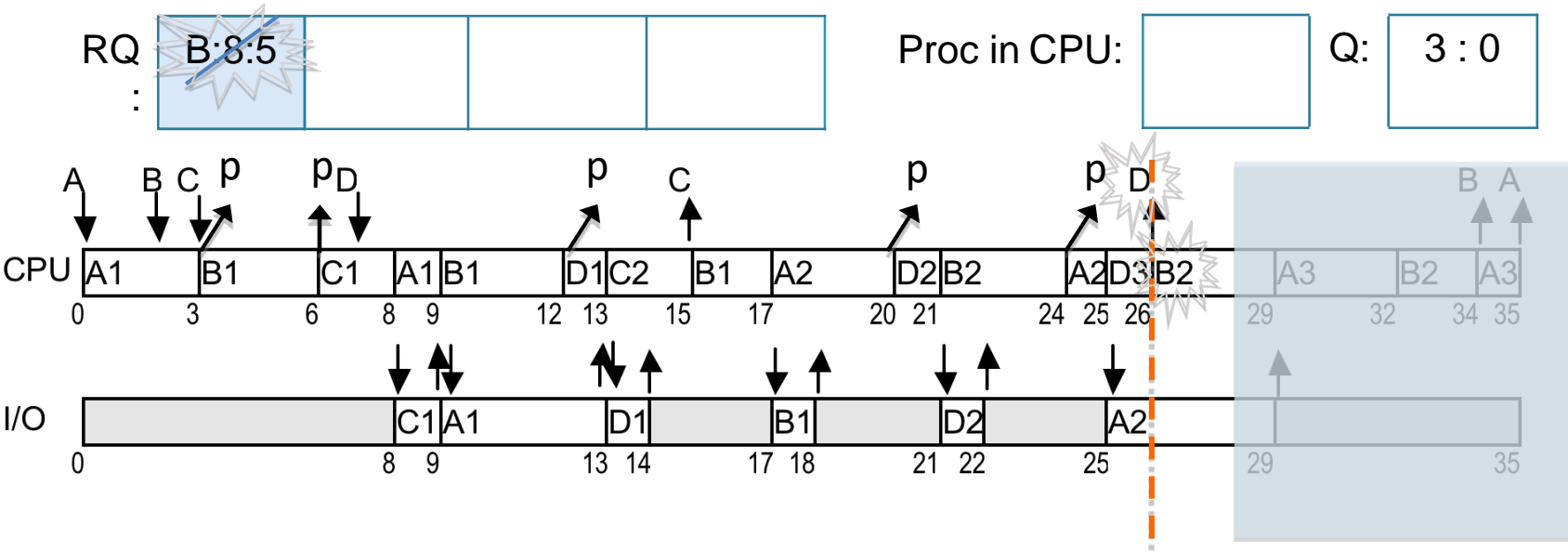
Q:

3 : 0



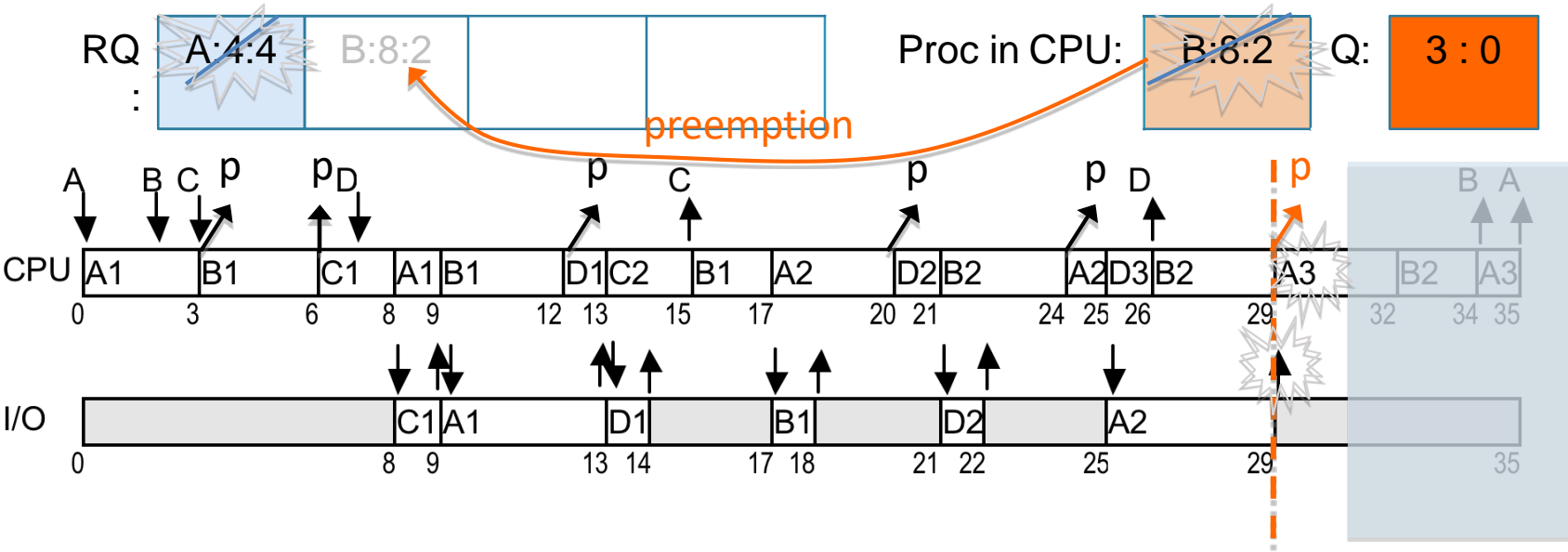
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 4
B	2	8 : 0	1	8 : 5	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0



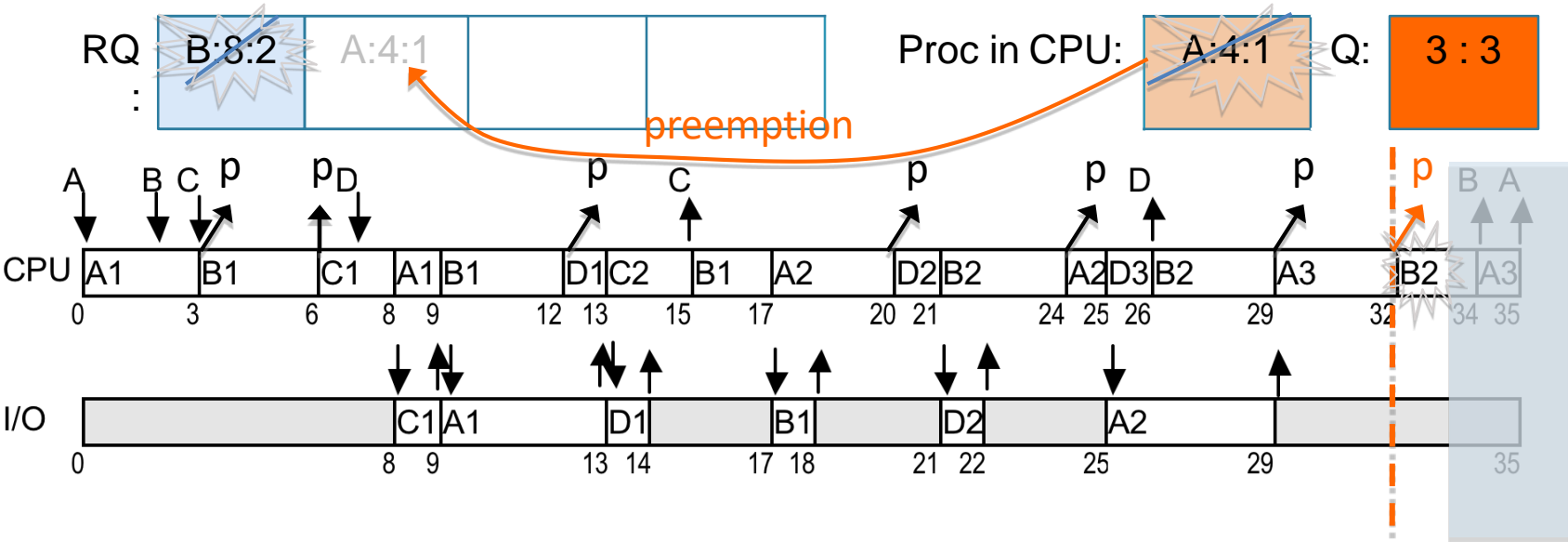
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 4
B	2	8 : 0	1	8 : 2	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0



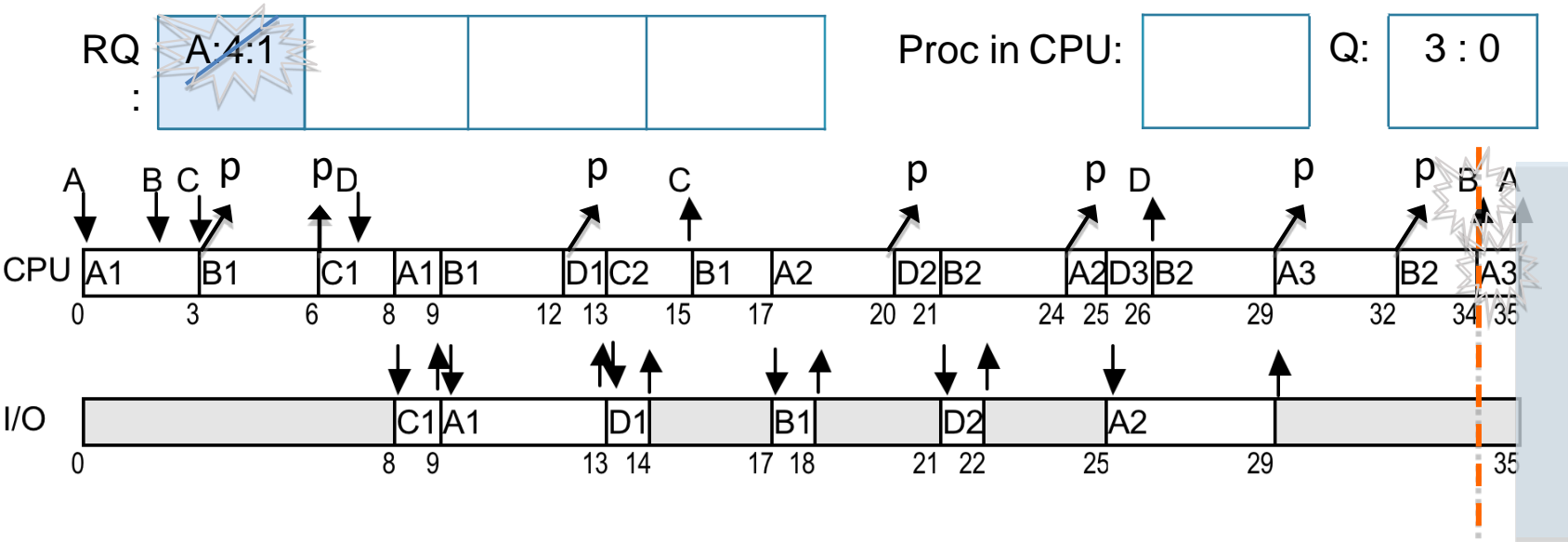
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 1
B	2	8 : 0	1	8 : 2	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0



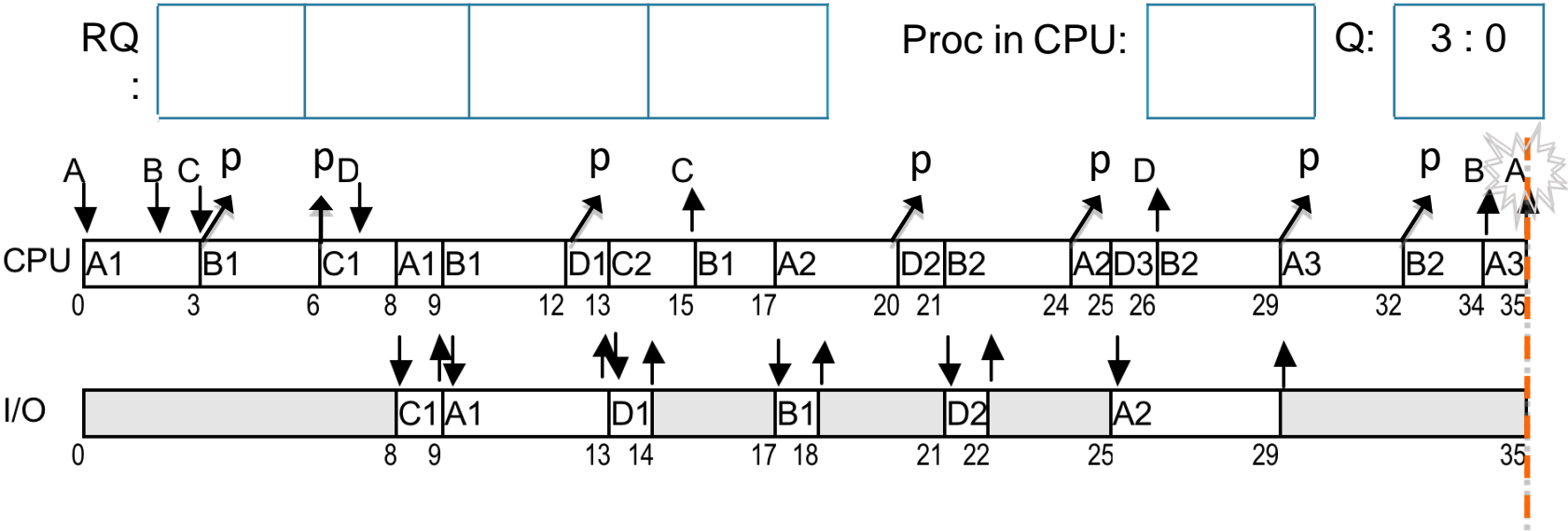
# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 1
B	2	8 : 0	1	8 : 0	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0

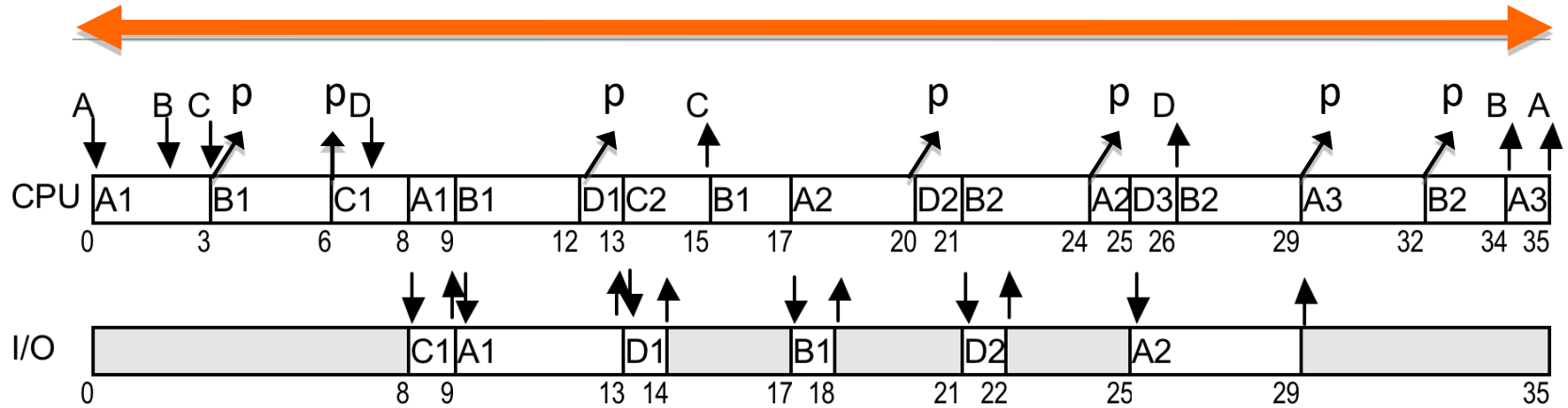


# Algorithme du Tourniquet (Round Robin)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 0
B	2	8 : 0	1	8 : 0	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0

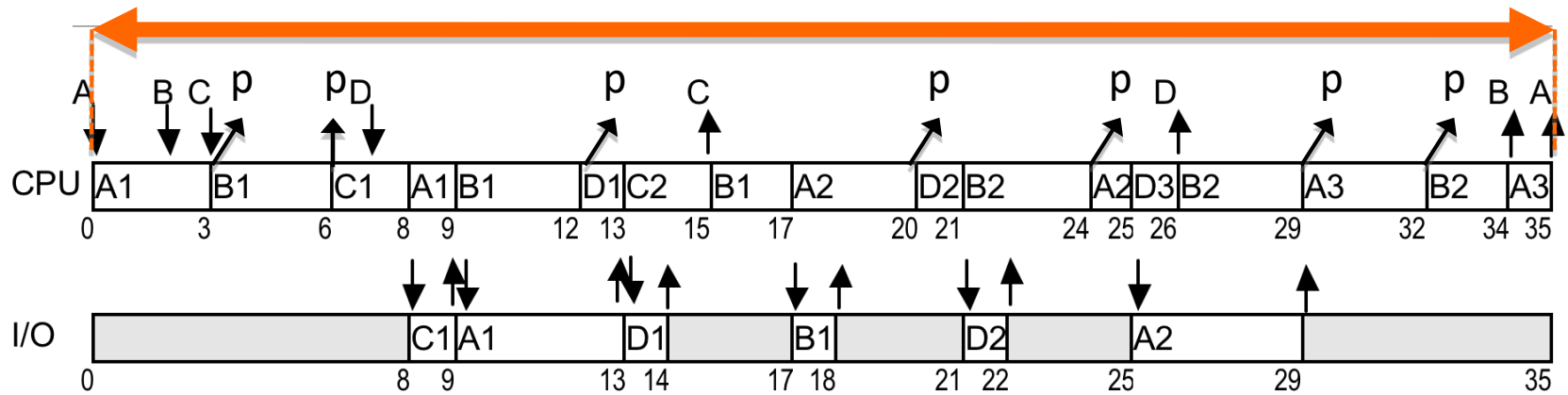


# Algorithme du Tourniquet (Round Robin)



- Processor utilization =  $(35 / 35) * 100 = 100 \%$
- Throughput =  $4 / 35$

# Algorithme du Tourniquet (Round Robin)

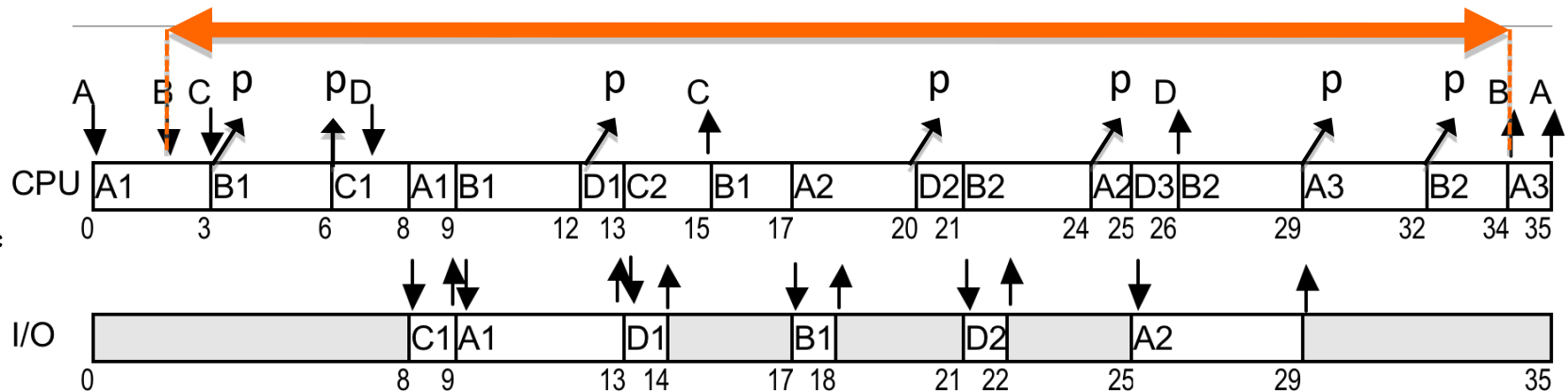


- Turn around time:  
 $tat_A = 35 - 0 = 35$   
 $tat_B = 34 - 2 = 32$   
 $tat_C = 15 - 3 = 12$   
 $tat_D = 26 - 7 = 19$

$$tat_{AVG} = (35 + 32 + 12 + 19) / 4 = 24.$$



# Algorithme du Tourniquet (Round Robin)



## Turn around time:

$$tat_A = 35 - 0 = 35$$

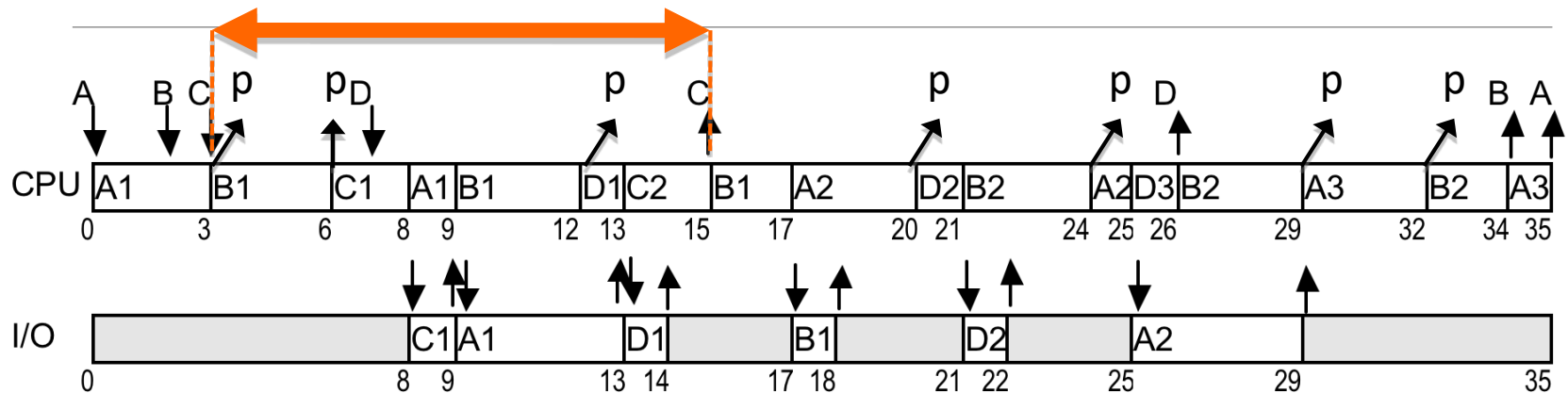
$$tat_B = 34 - 2 = 32$$

$$tat_C = 15 - 3 = 12$$

$$tat_D = 26 - 7 = 19$$

$$tat_{AVG} = (35 + 32 + 12 + 19) / 4 = 24.$$

# Algorithme du Tourniquet (Round Robin)



- Turn around time:

$$tat_A = 35 - 0 = 35$$

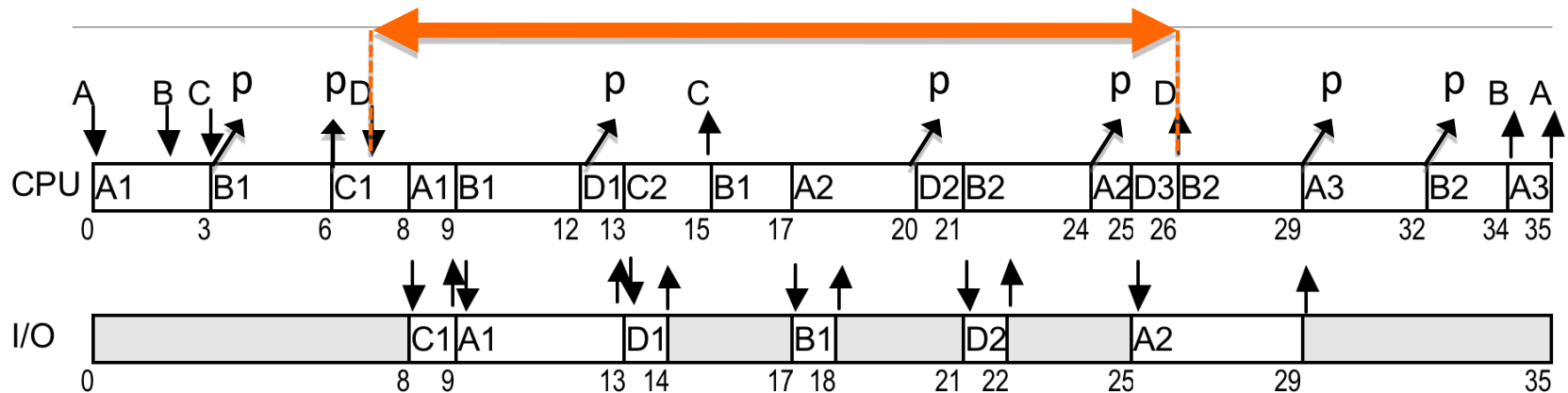
$$tat_B = 34 - 2 = 32$$

$$tat_C = 15 - 3 = 12$$

$$tat_D = 26 - 7 = 19$$

$$tat_{AVG} = (35 + 32 + 12 + 19) / 4 = 24.$$

# Algorithme du Tourniquet (Round Robin)



- Turn around time:

$$tat_A = 35 - 0 = 35$$

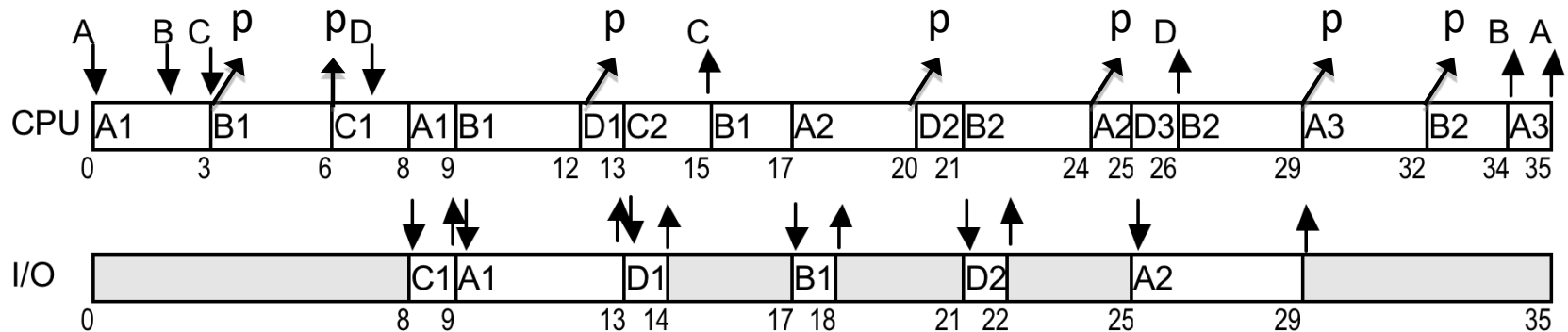
$$tat_B = 34 - 2 = 32$$

$$tat_C = 15 - 3 = 12$$

$$tat_D = 26 - 7 = 19$$

$$tat_{AVG} = (35 + 32 + 12 + 19) / 4 = 24.$$

# Algorithme du Tourniquet (Round Robin)



- Turn around time:

$$tat_A = 35 - 0 = 35$$

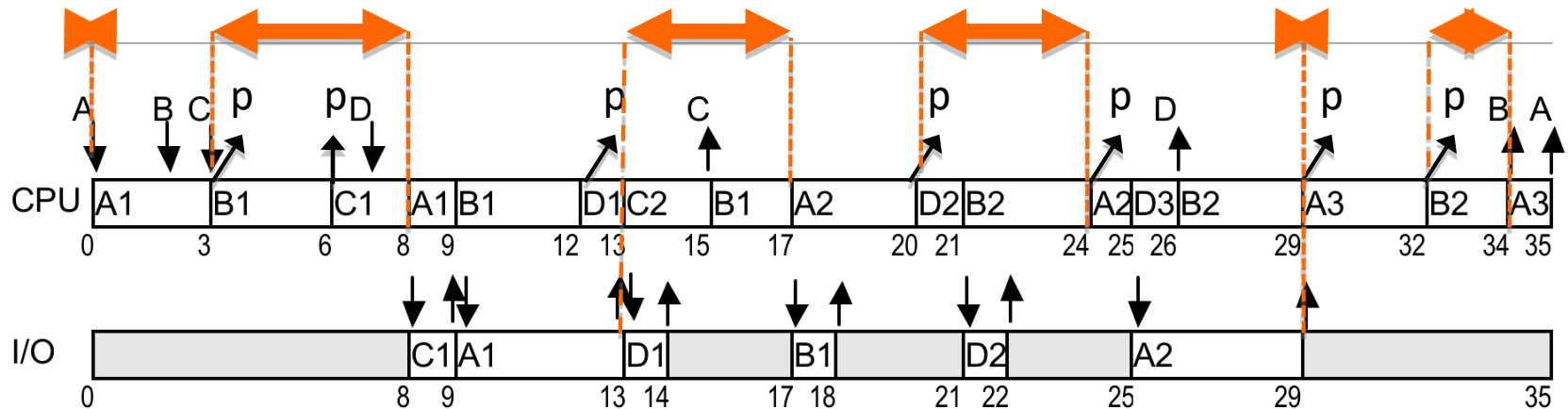
$$tat_B = 34 - 2 = 32$$

$$tat_C = 15 - 3 = 12$$

$$tat_D = 26 - 7 = 19$$

$$tat_{AVG} = (35 + 32 + 12 + 19) / 4 = 24$$

# Algorithme du Tourniquet (Round Robin)



- Waiting time:

$$wt_A = (0 - 0) + (8 - 3) + (17 - 13) + (24 - 20) + (29 - 29) + (34 - 32) = 15$$

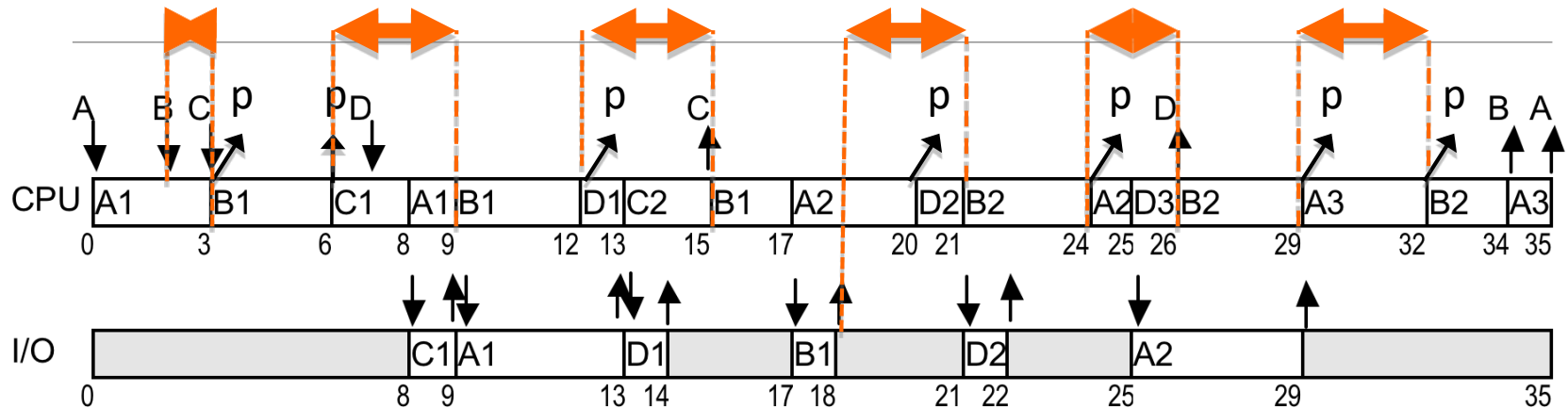
$$wt_B = (3 - 2) + (9 - 6) + (15 - 12) + (21 - 18) + (26 - 24) + (32 - 29) = 15$$

$$wt_C = (6 - 3) + (13 - 9) = 7$$

$$wt_D = (12 - 7) + (20 - 14) + (25 - 22) = 14$$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$

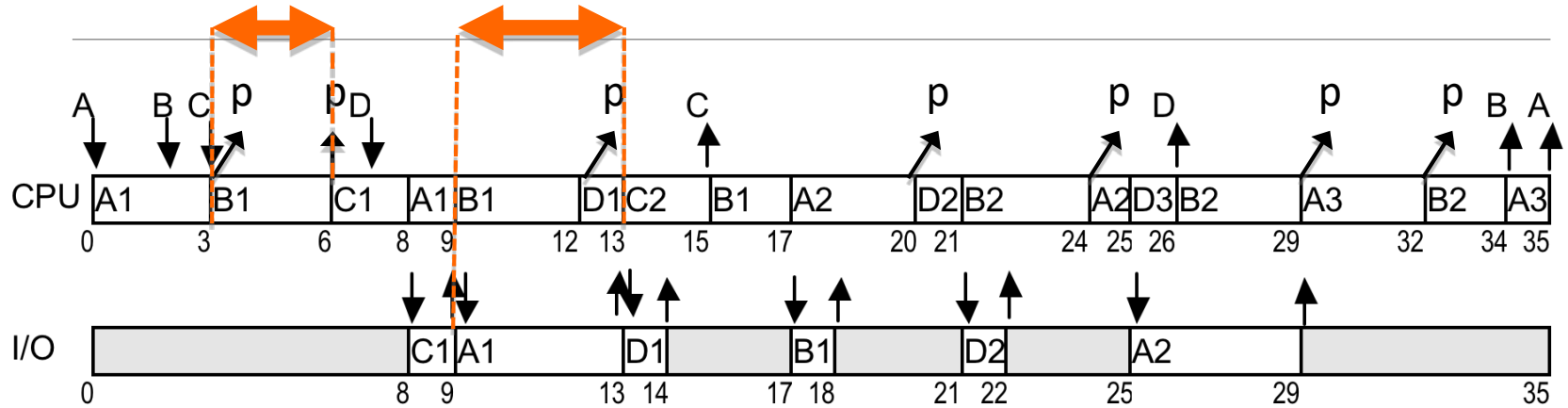
# Algorithme du Tourniquet (Round Robin)



- Waiting time:
  - $wt_A = (0 - 0) + (8 - 3) + (17 - 13) + (24 - 20) + (29 - 29) + (34 - 32) = 15$
  - $wt_B = (3 - 2) + (9 - 6) + (15 - 12) + (21 - 18) + (26 - 24) + (32 - 29) = 15$
  - $wt_C = (6 - 3) + (13 - 9) = 7$
  - $wt_D = (12 - 7) + (20 - 14) + (25 - 22) = 14$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$

# Algorithme du Tourniquet (Round Robin)



- Waiting time:

$$wt_A = (0 - 0) + (8 - 3) + (17 - 13) + (24 - 20) + (29 - 29) + (34 - 32) = 15$$

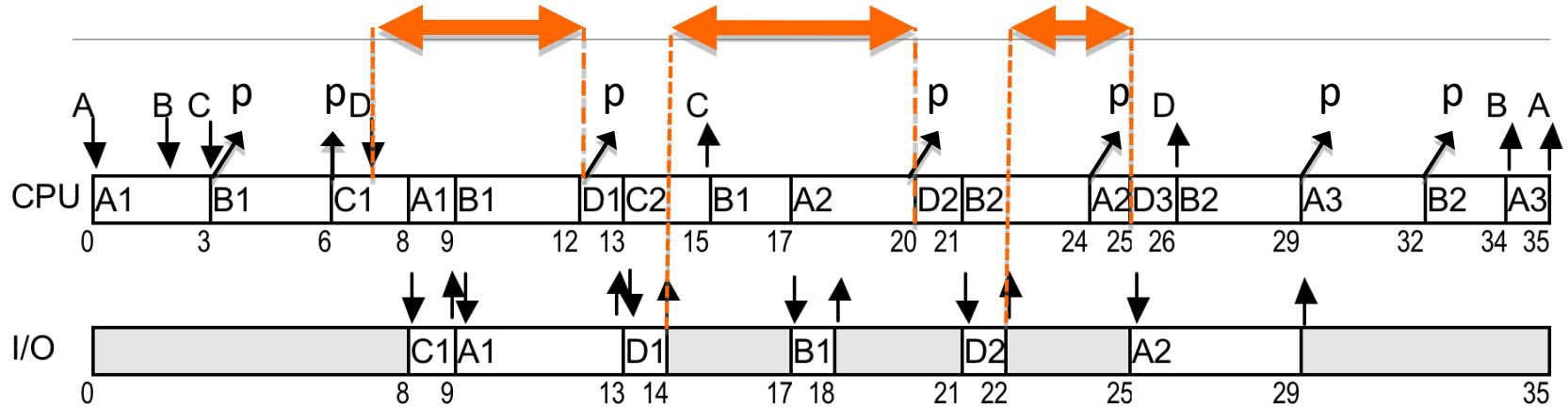
$$wt_B = (3 - 2) + (9 - 6) + (15 - 12) + (21 - 18) + (26 - 24) + (32 - 29) = 15$$

$$wt_C = (6 - 3) + (13 - 9) = 7$$

$$wt_D = (12 - 7) + (20 - 14) + (25 - 22) = 14$$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$

# Algorithme du Tourniquet (Round Robin)

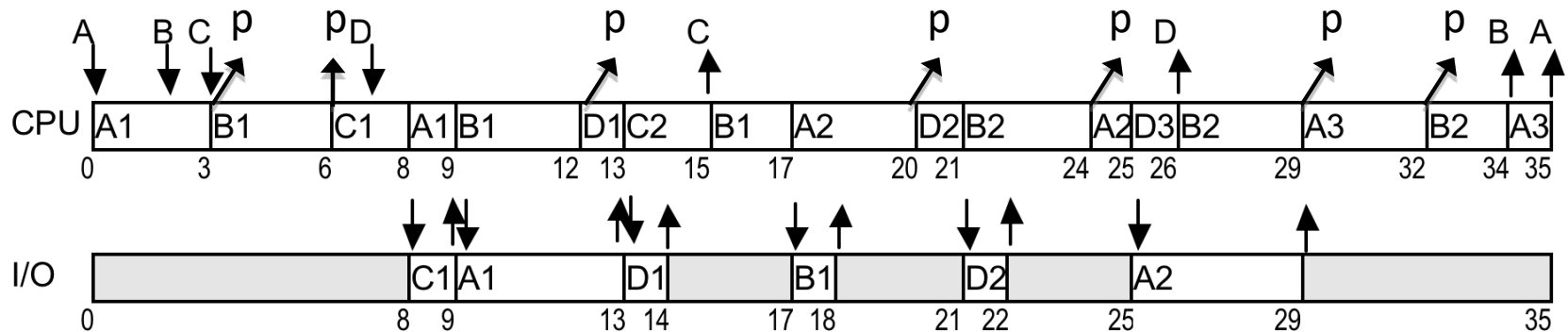


- Waiting time:
  - $wt_A = (0 - 0) + (8 - 3) + (17 - 13) + (24 - 20) + (29 - 29) + (34 - 32) = 15$
  - $wt_B = (3 - 2) + (9 - 6) + (15 - 12) + (21 - 18) + (26 - 24) + (32 - 29) = 15$
  - $wt_C = (6 - 3) + (13 - 9) = 7$
  - $wt_D = (12 - 7) + (20 - 14) + (25 - 22) = 14$

$$wt_{AVG} = (15 + 12 + 7 + 11) / 4 = 11.25$$



# Algorithme du Tourniquet (Round Robin)



- Waiting time:

$$wt_A = (0 - 0) + (8 - 3) + (17 - 13) + (24 - 20) + (29 - 29) + (34 - 32) = 15$$

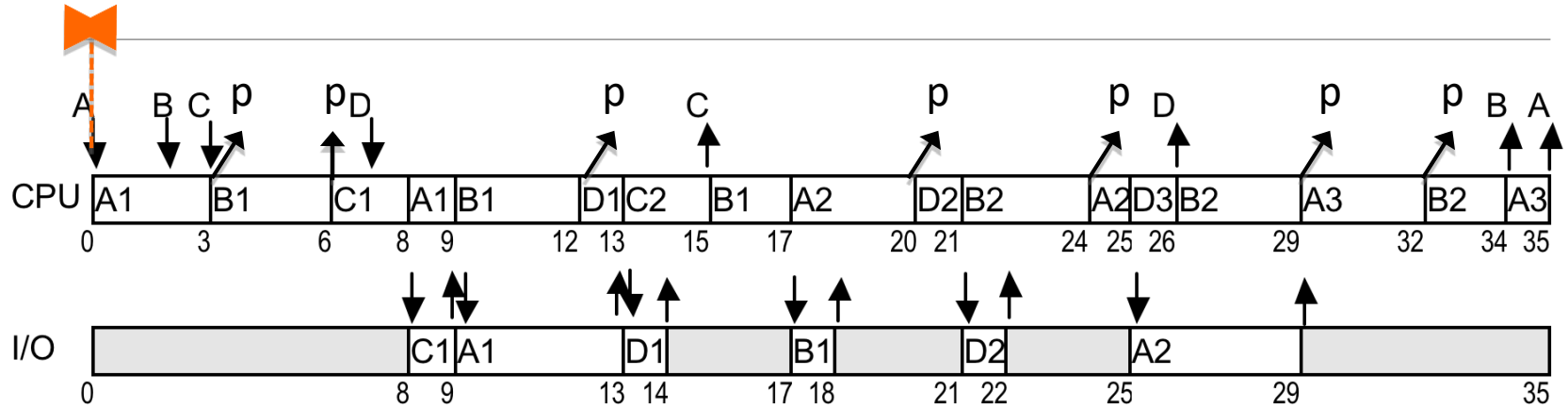
$$wt_B = (3 - 2) + (9 - 6) + (15 - 12) + (21 - 18) + (26 - 24) + (32 - 29) = 15$$

$$wt_C = (6 - 3) + (13 - 9) = 7$$

$$wt_D = (12 - 7) + (20 - 14) + (25 - 22) = 14$$

$$wt_{AVG} = (15 + 14 + 7 + 11) / 4 = 12.75$$

# Algorithme du Tourniquet (Round Robin)



• Response time:

$$rt_A = 0 - 0 = 0$$

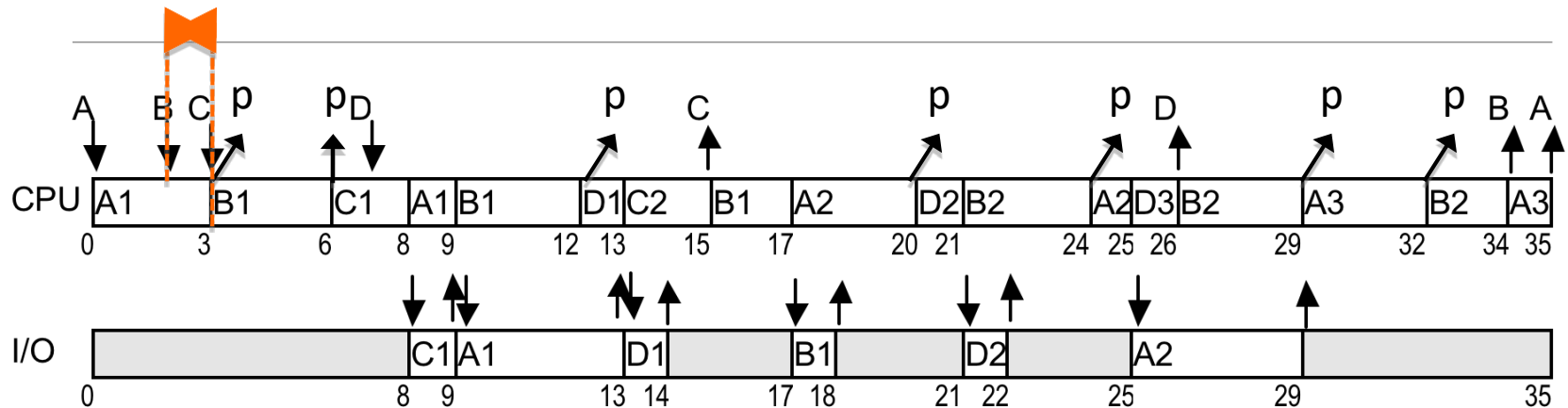
$$rt_B = 36 - 2 = 1$$

$$rt_C = 6 - 3 = 3$$

$$rt_D = 12 - 7 = 5$$

$$rt_{AVG} = (0 + 1 + 3 + 5) / 4 = 2.25$$

# Algorithme du Tourniquet (Round Robin)



- Response time:

$$rt_A = 0 - 0 = 0$$

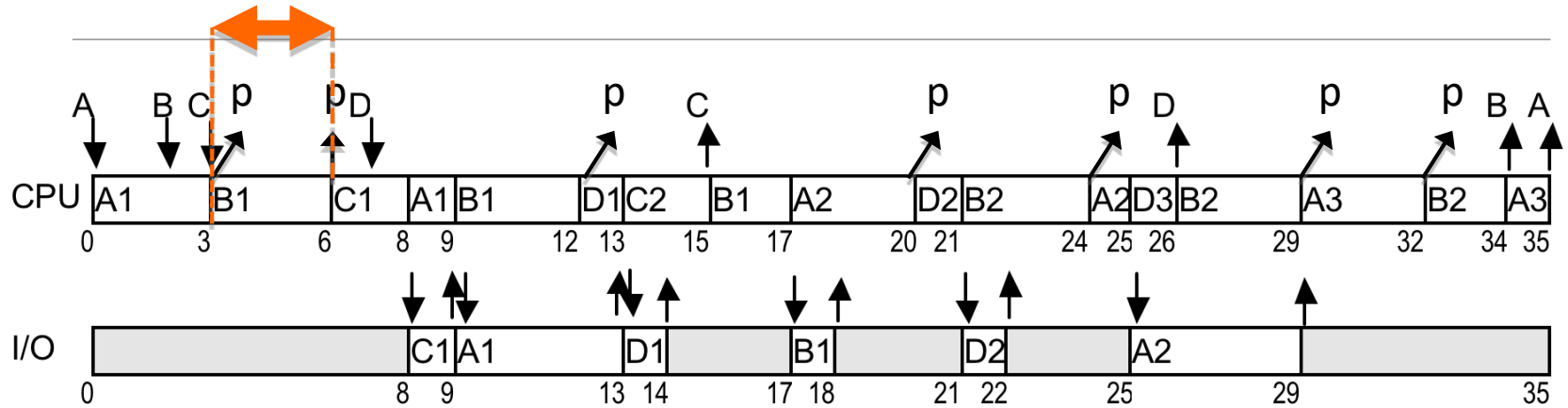
$$rt_B = 3 - 2 = 1$$

$$rt_C = 6 - 3 = 3$$

$$rt_D = 12 - 7 = 5$$

$$rt_{AVG} = (0 + 1 + 3 + 5) / 4 = 2.25$$

# Algorithme du Tourniquet (Round Robin)



- Response time:

$$rt_A = 0 - 0 = 0$$

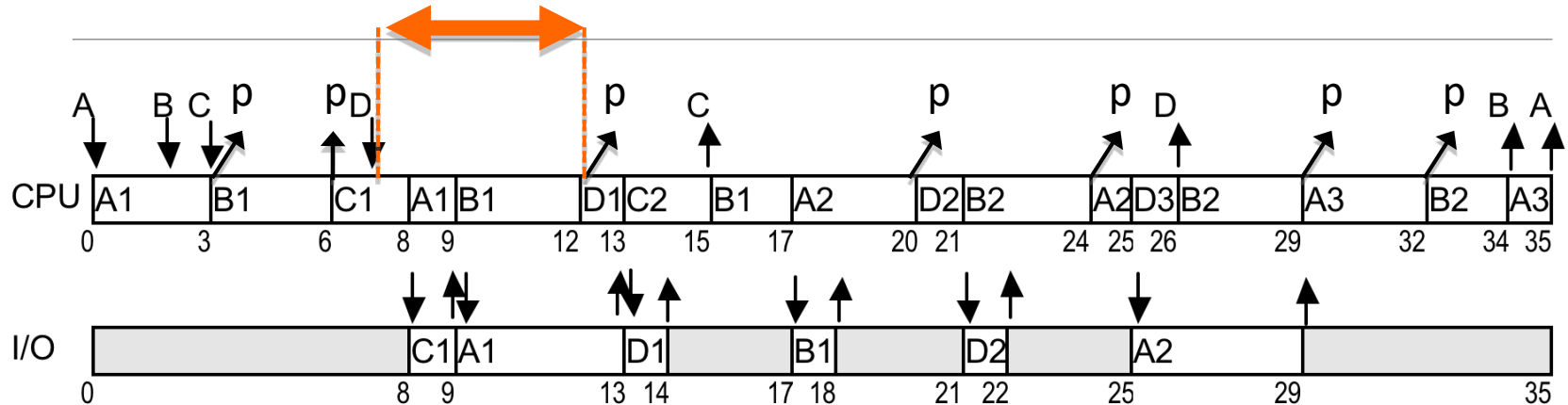
$$rt_B = 36 - 2 = 1$$

$$rt_C = 6 - 3 = 3$$

$$rt_D = 12 - 7 = 5$$

$$rt_{AVG} = (0 + 1 + 3 + 5) / 4 = 2.25$$

# Algorithme du Tourniquet (Round Robin)



- Response time:

$$rt_A = 0 - 0 = 0$$

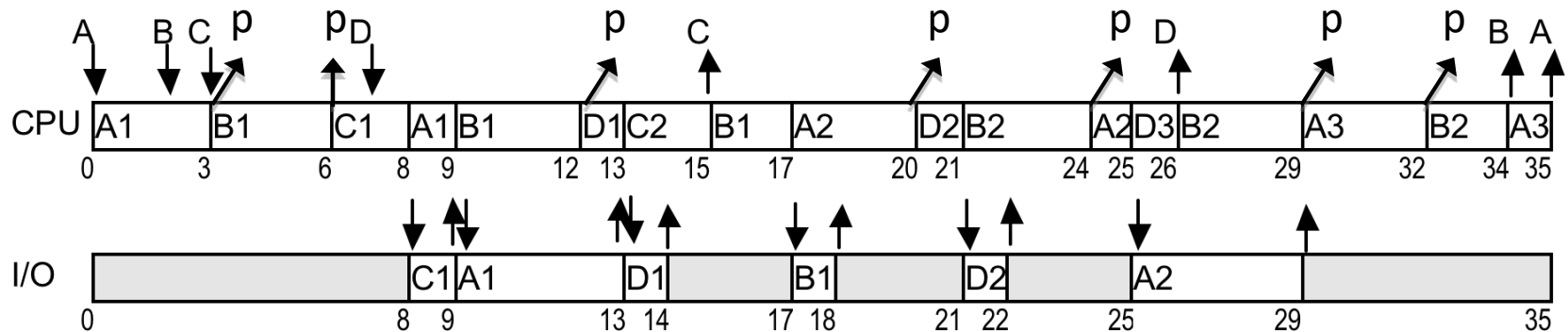
$$rt_B = 3 - 2 = 1$$

$$rt_C = 6 - 3 = 3$$

$$rt_D = 12 - 7 = 5$$

$$rt_{AVG} = (0 + 1 + 3 + 5) / 4 = 2.25$$

# Algorithme du Tourniquet (Round Robin)



- Response time:

$$rt_A = 0 - 0 = 0$$

$$rt_B = 3 - 2 = 1$$

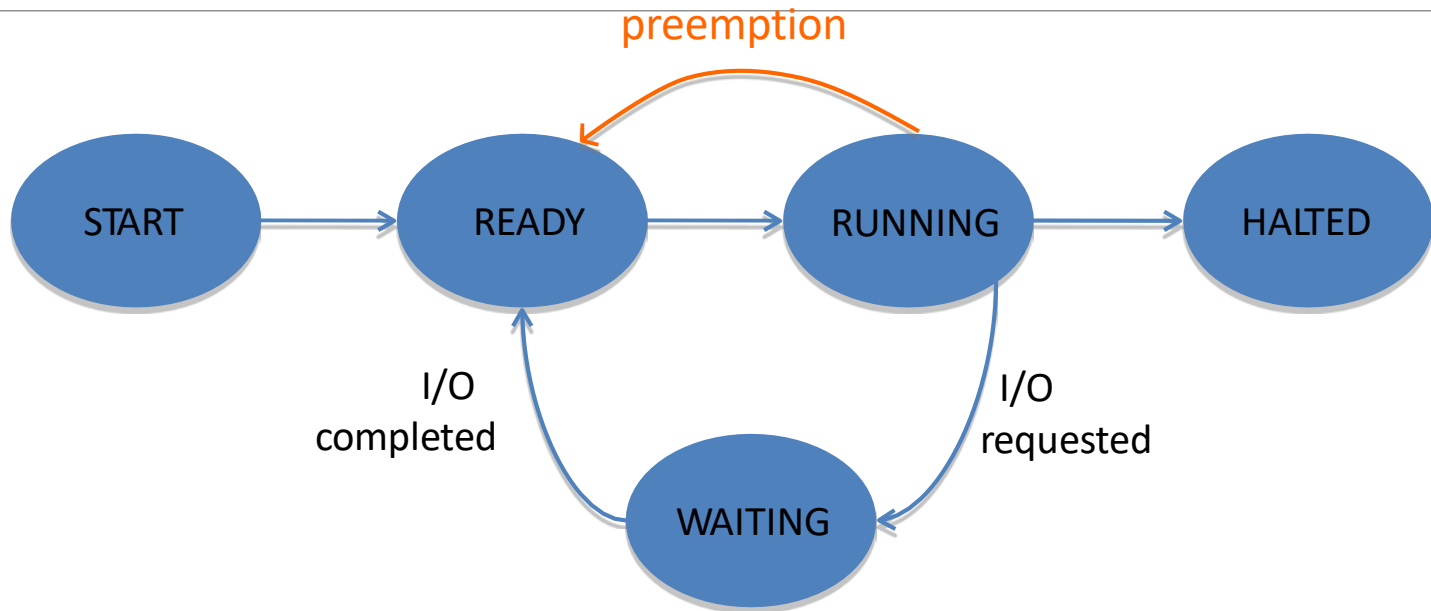
$$rt_C = 6 - 3 = 3$$

$$rt_D = 12 - 7 = 5$$

$$rt_{AVG} = (0 + 1 + 3 + 5) / 4 = 2.25$$

PLUS COURT TEMPS  
D'UTILISATION  
D'ABORD PRÉEMPTIF

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



Start : Le processus vient d'être créé.

Ready : Le processus attend d'obtenir le processeur.

Running : Le processus a été alloué par le processeur.

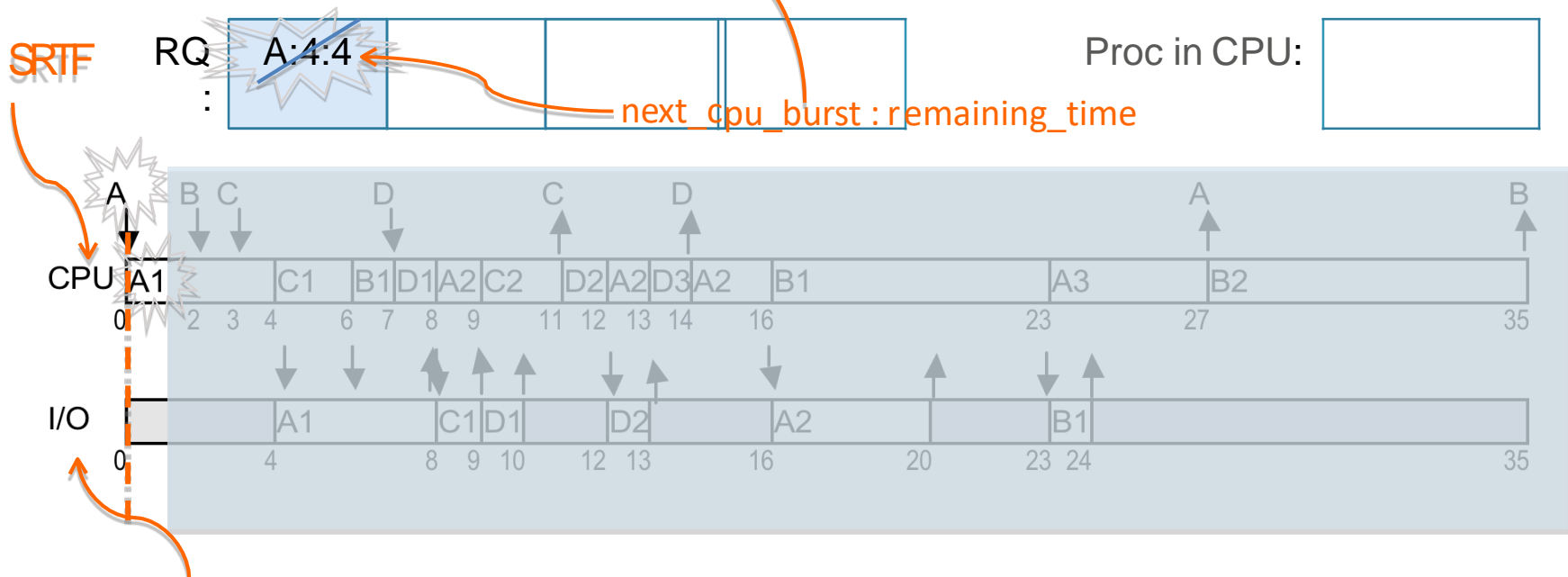
Waiting : Le processeur est entré en E/S ou bloqué.

Halted : Le processus a fini et va quitter le système



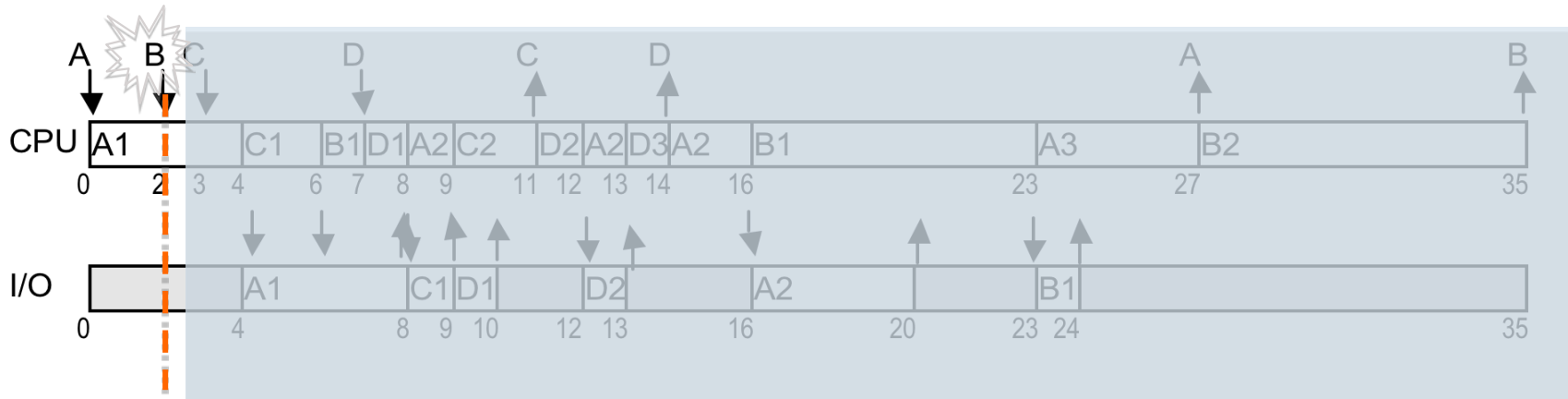
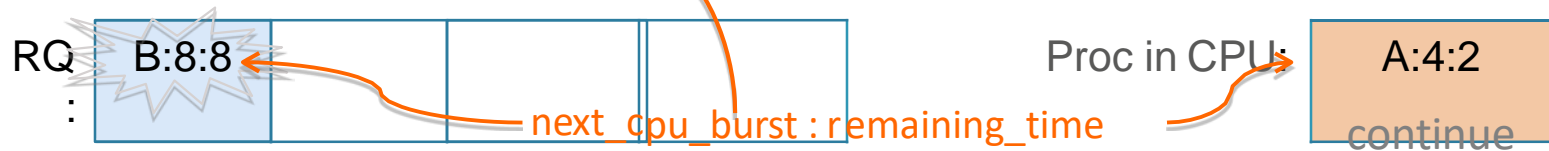
## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 4	4	4 : 4	4	4 : 4
B	2	8 : 8	1	8 : 8	-	-
C	3	2 : 2	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



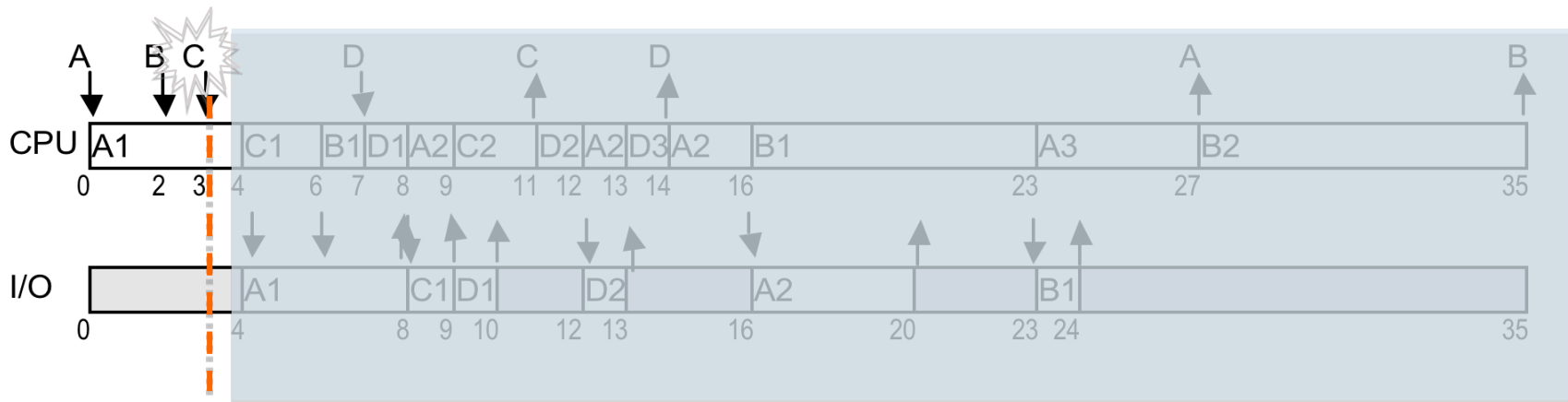
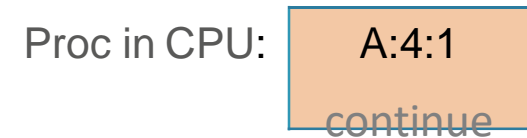
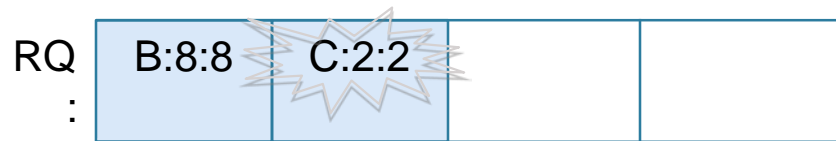
## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 2	4	4 : 4	4	4 : 4
B	2	8 : 8	1	8 : 8	-	-
C	3	2 : 2	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 1	4	4 : 4	4	4 : 4
B	2	8 : 8	1	8 : 8	-	-
C	3	2 : 2	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1



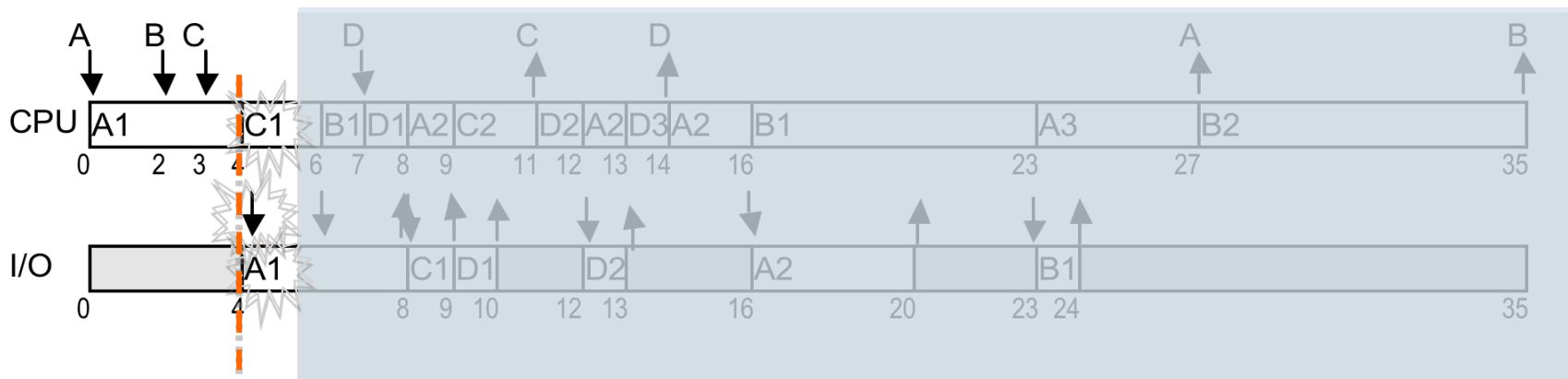
## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 8	1	8 : 8	-	-
C	3	2 : 2	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1

RQ : 

B:8:8	<del>C:2:2</del>		
-------	------------------	--	--

Proc in CPU:

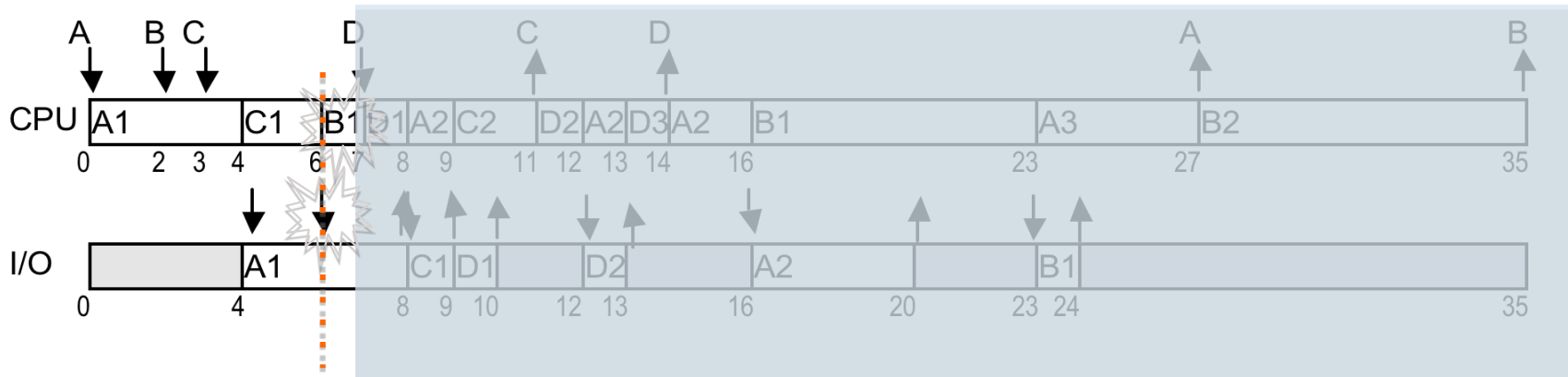


## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 8	1	8 : 8	-	-
C	3	2 : 0	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1

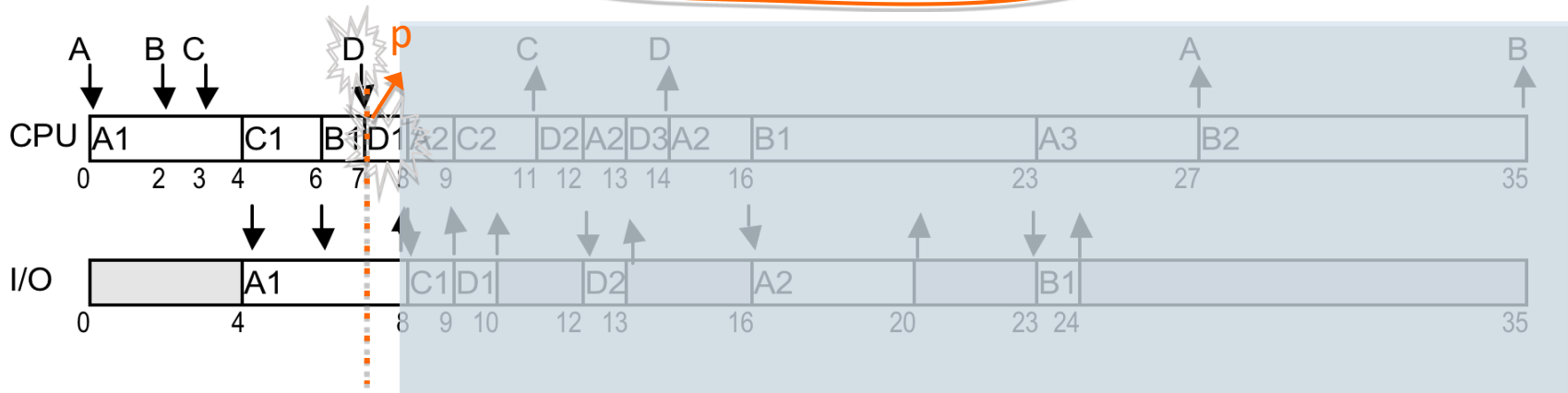
RQ : B:8.8

Proc in CPU:



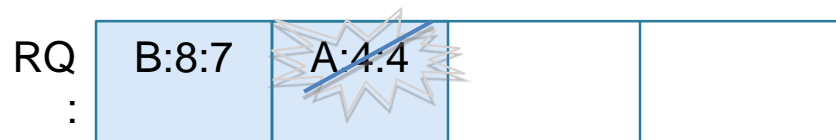
## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 2	-	-
D	7	1 : 1	1	1 : 1	1	1 : 1

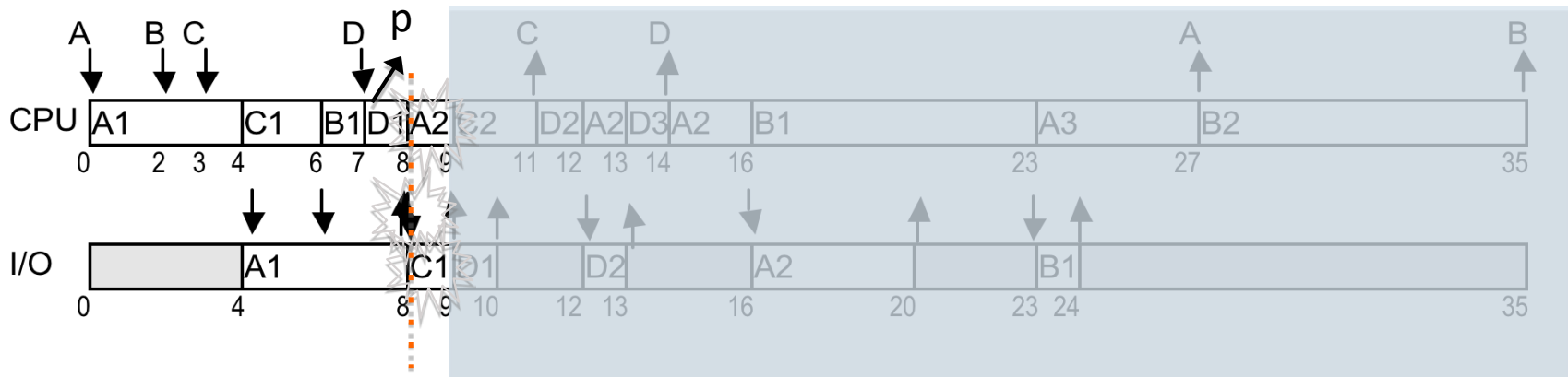
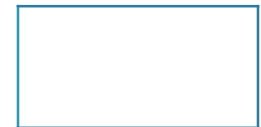


## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 4	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 2	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1

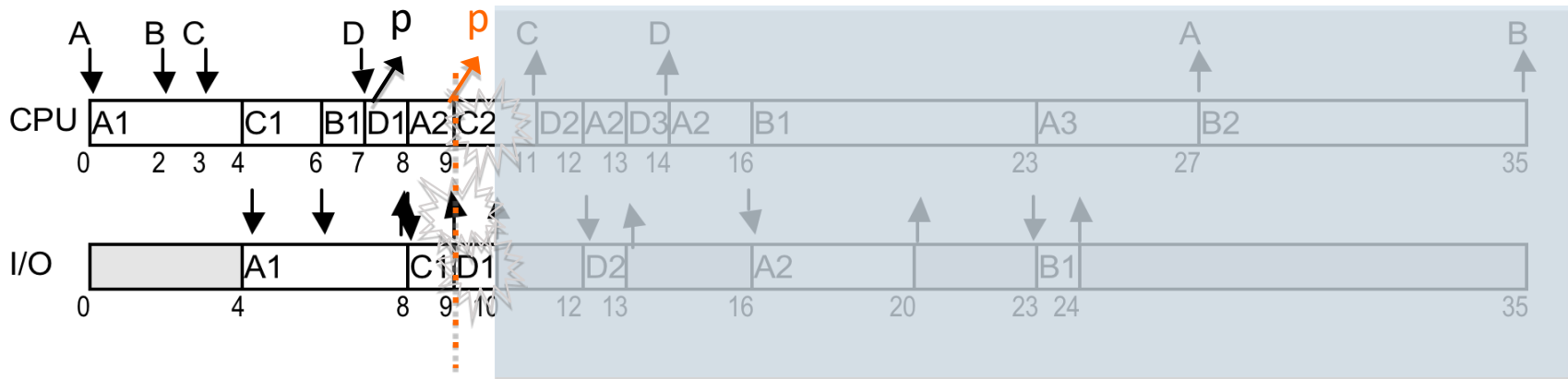


Proc in CPU:



## 2.3.3 Shortest-Remaining-Time-First (SRTF)

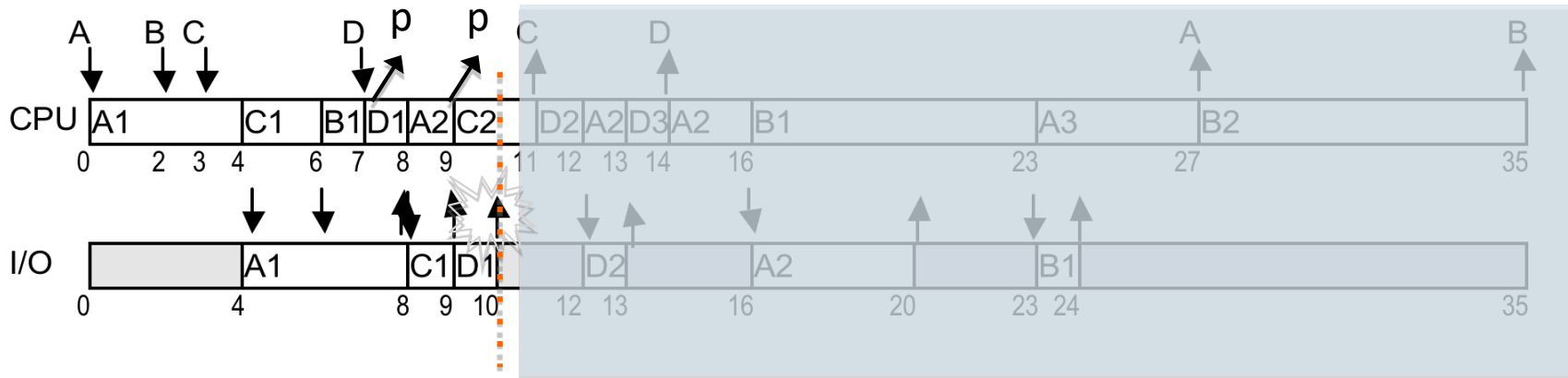
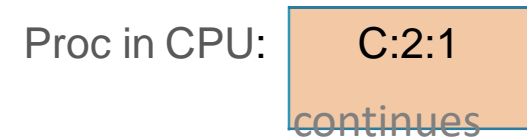
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 3	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 2	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1





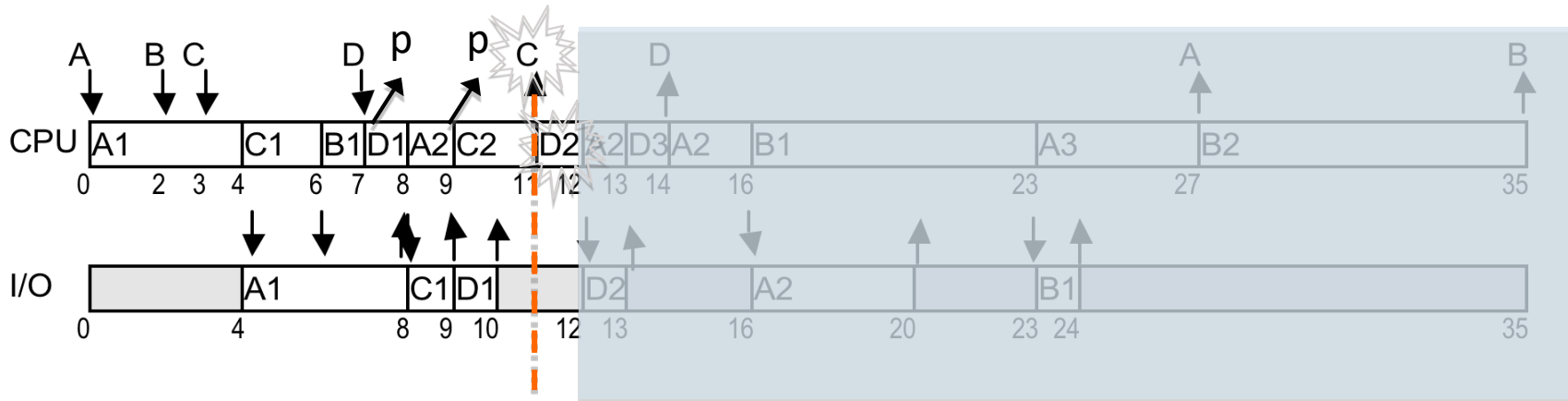
## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 3	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 1	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1



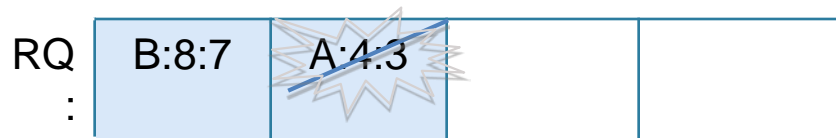
## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 3	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 1	1	1 : 1

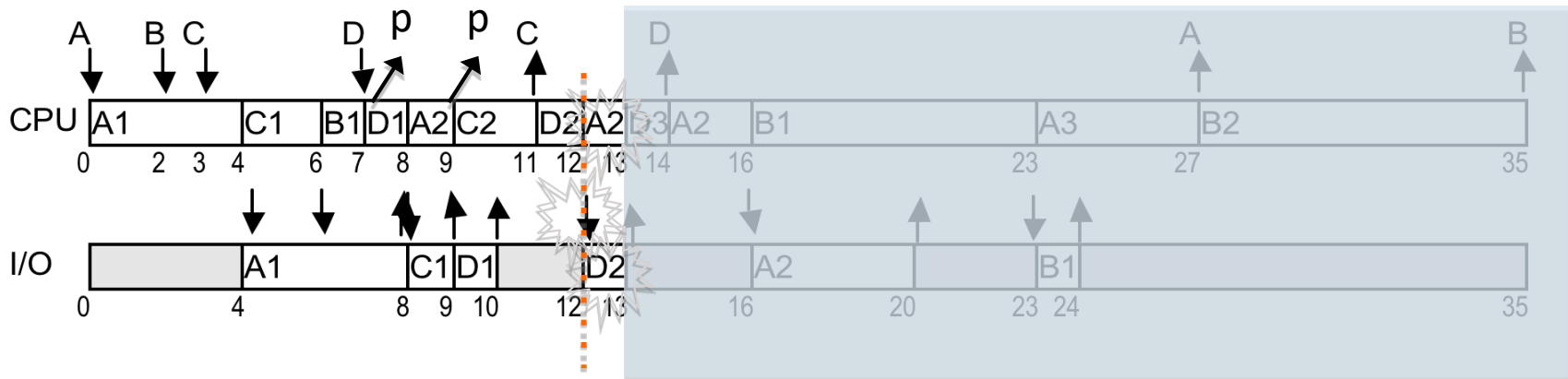
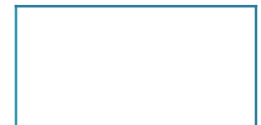


## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 3	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 1

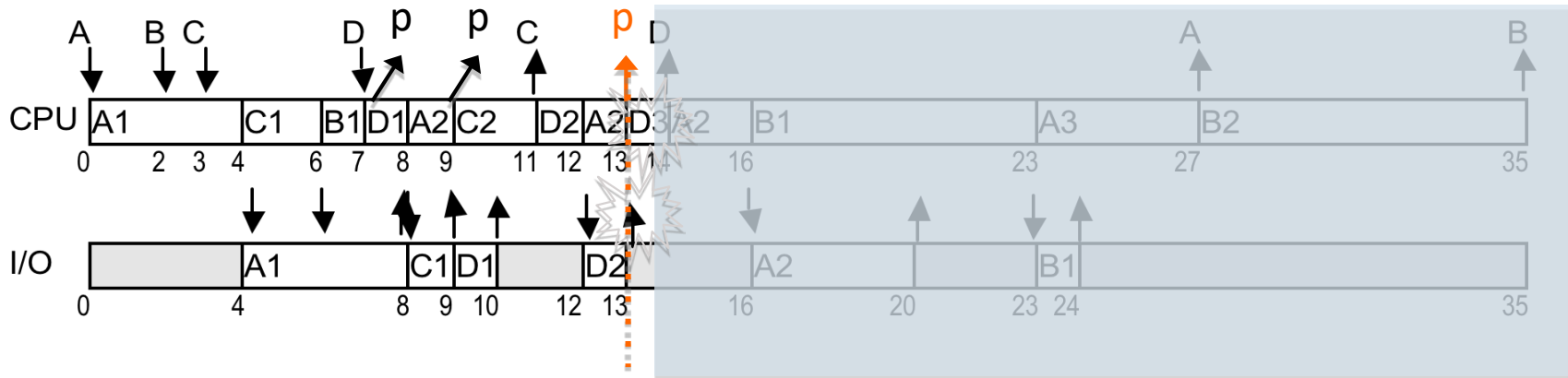


Proc in CPU:



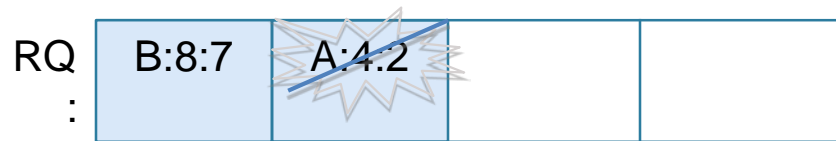
## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 2	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 1

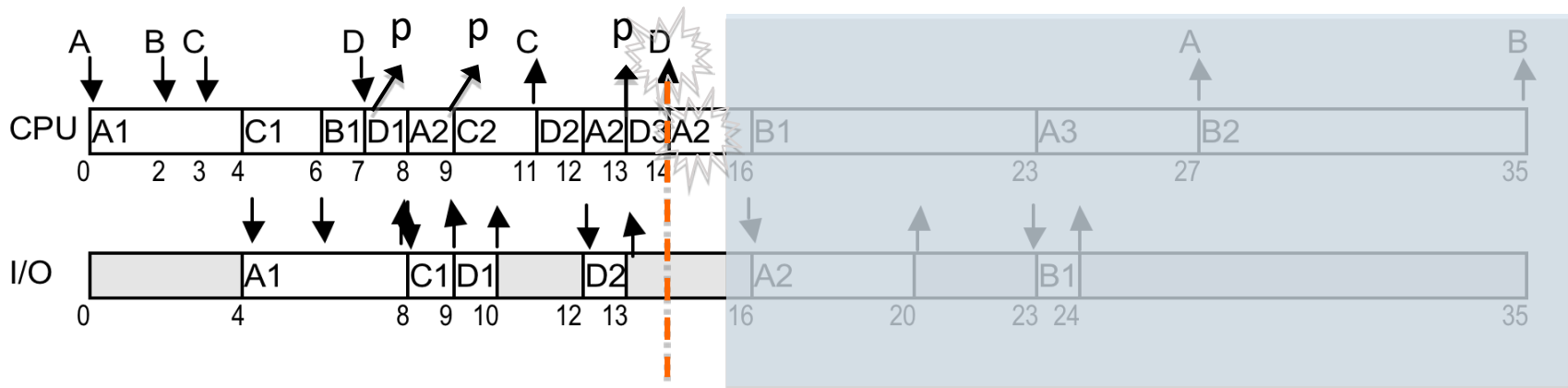
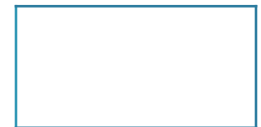


## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 2	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0



Proc in CPU:

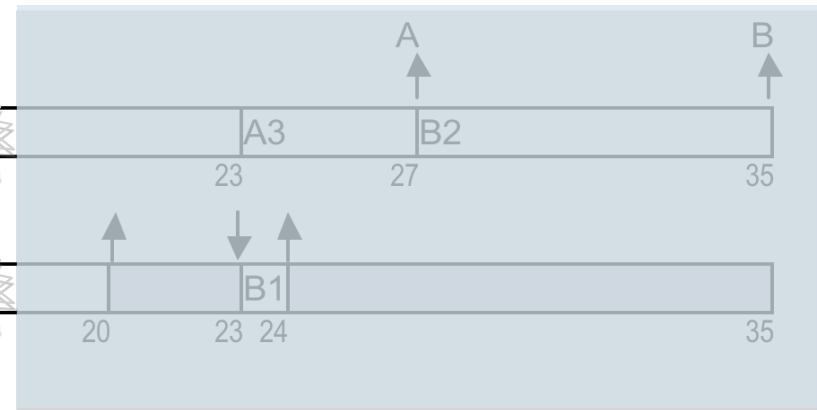
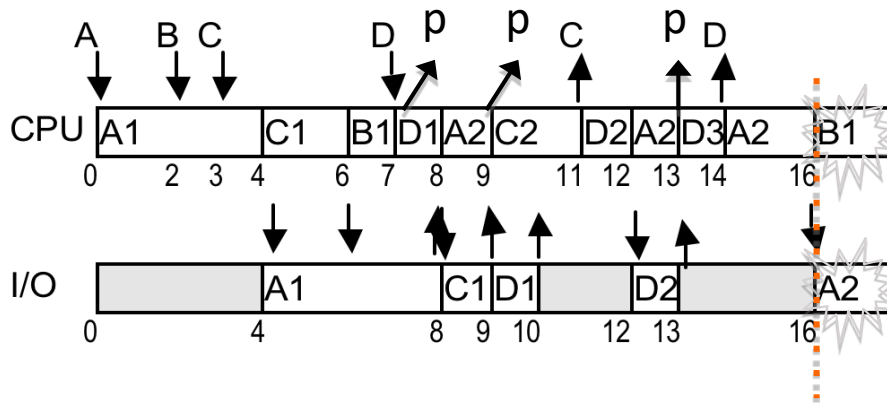


## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 4
B	2	8 : 7	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0

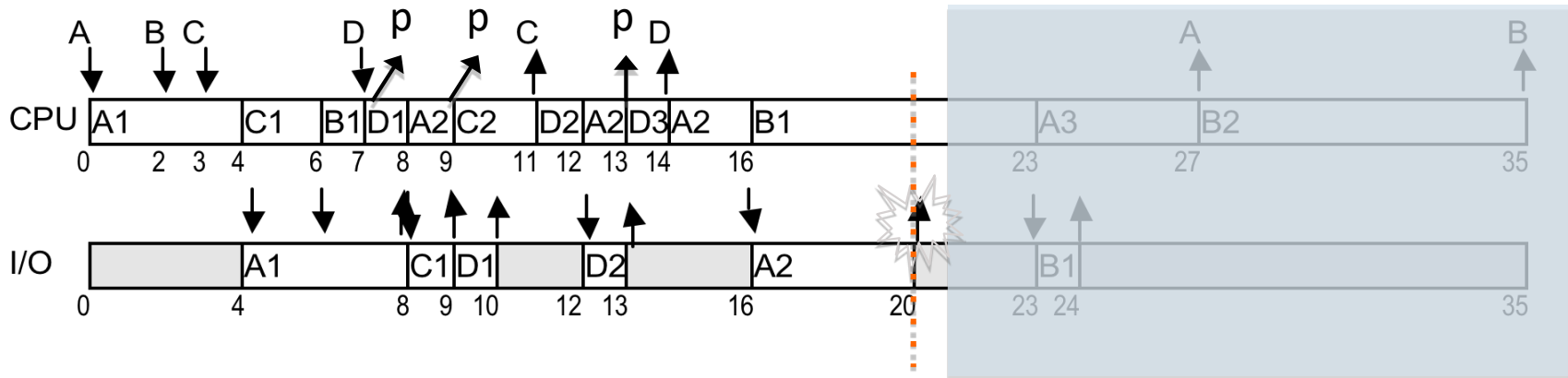
RQ : B:8:7

Proc in CPU:



## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 4
B	2	8 : 3	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0

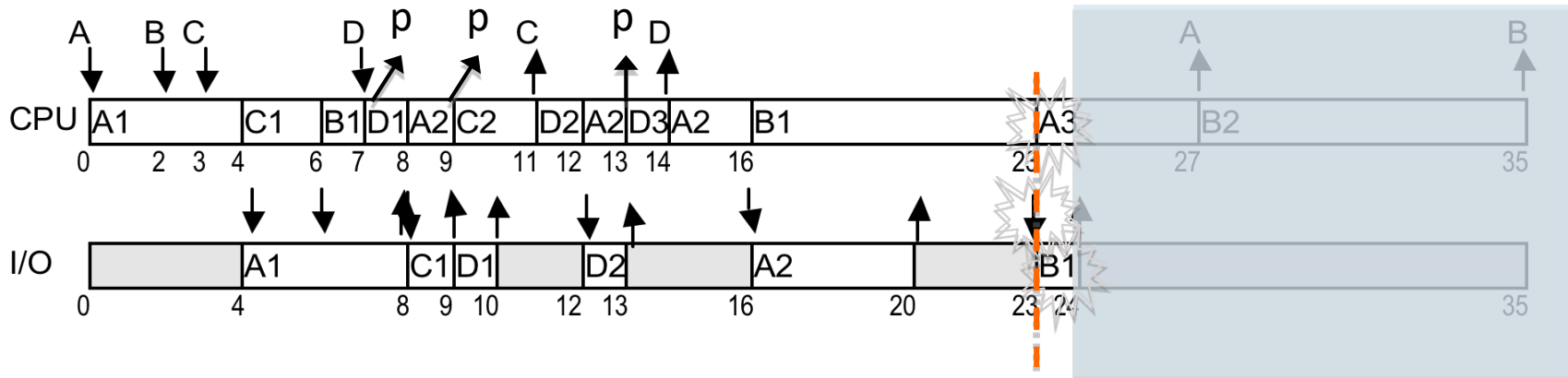


## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 4
B	2	8 : 0	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0

RQ : A:4:4

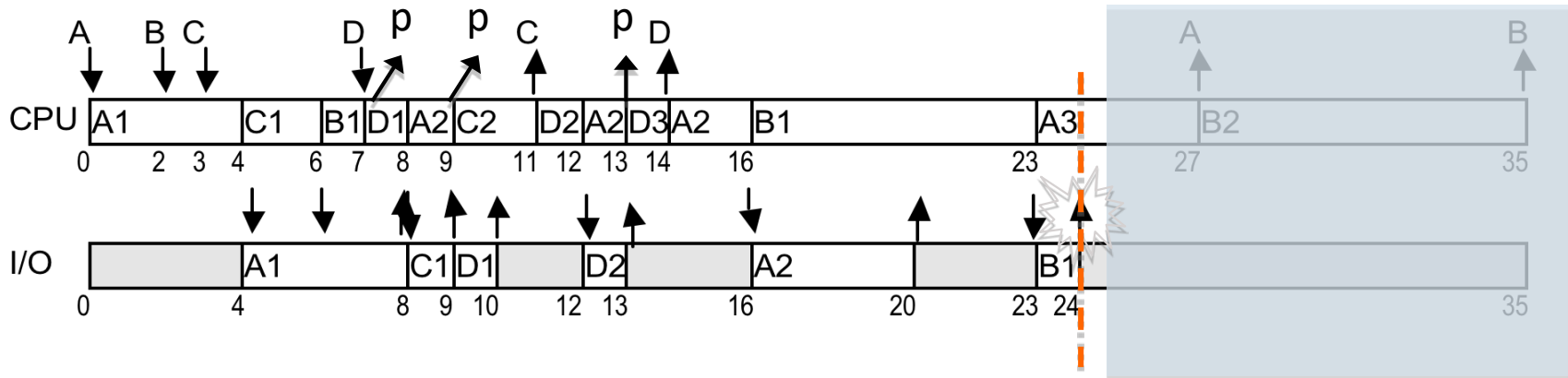
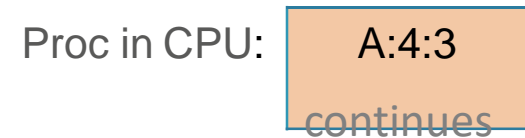
Proc in CPU:





## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 3
B	2	8 : 0	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0

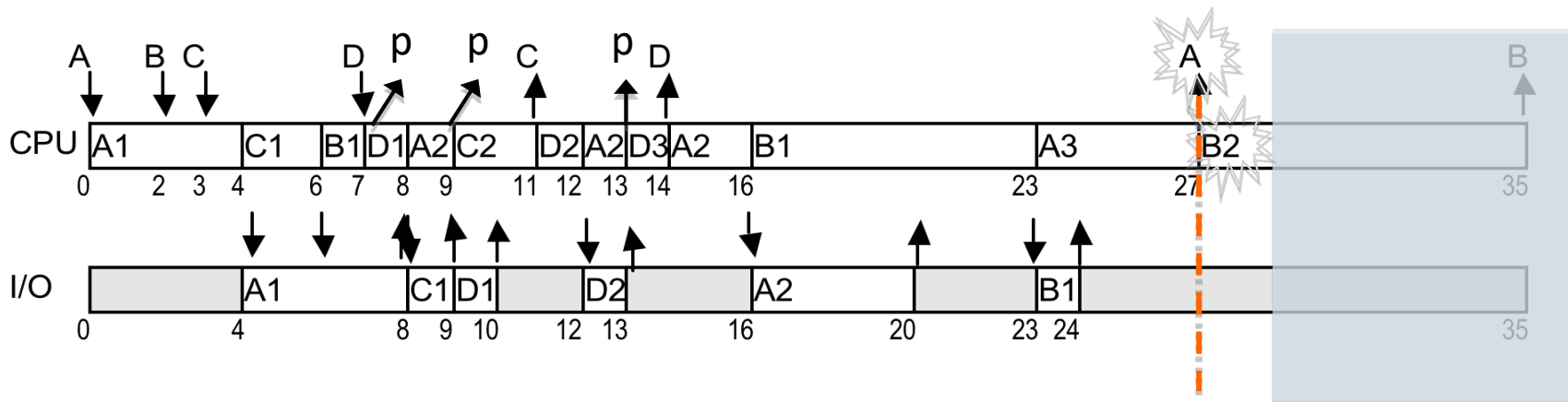


## 2.3.3 Shortest-Remaining-Time-First (SRTF)

Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 0
B	2	8 : 0	1	8 : 8	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0

RQ : B:8:8

Proc in CPU:



## 2.3.3 Shortest-Remaining-Time-First (SRTF)

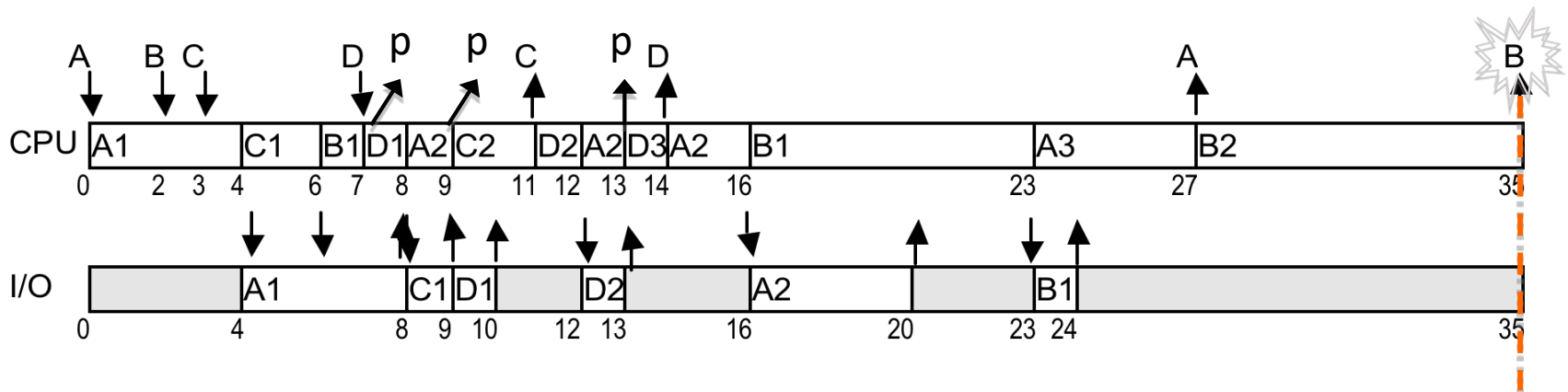
Process	Arrival time	1 <sup>st</sup> exec	1 <sup>st</sup> I/O	2 <sup>nd</sup> exec	2 <sup>nd</sup> I/O	3 <sup>rd</sup> exec
A	0	4 : 0	4	4 : 0	4	4 : 0
B	2	8 : 0	1	8 : 0	-	-
C	3	2 : 0	1	2 : 0	-	-
D	7	1 : 0	1	1 : 0	1	1 : 0

RQ  
:

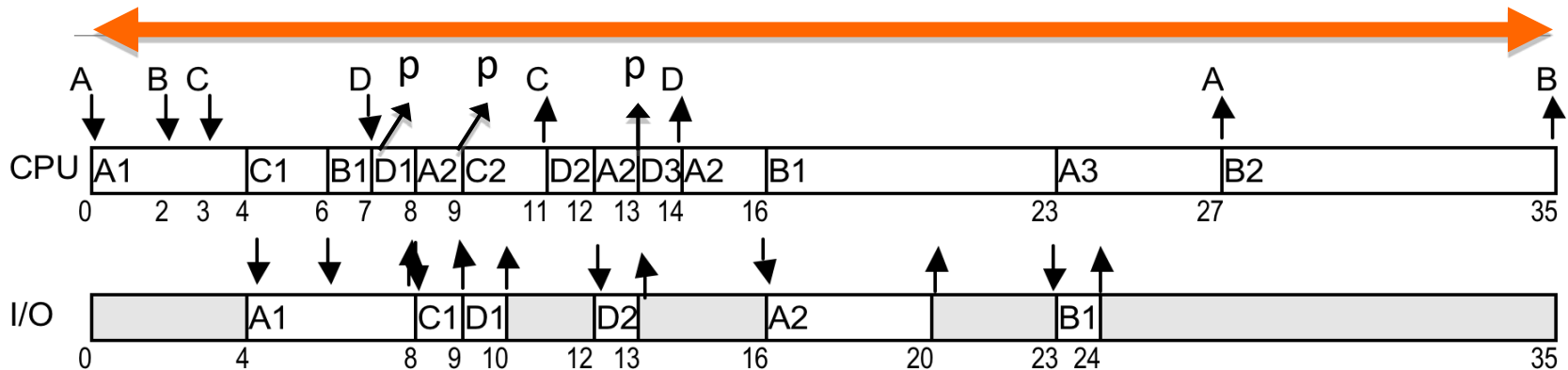
--	--	--	--

Proc in CPU:

--

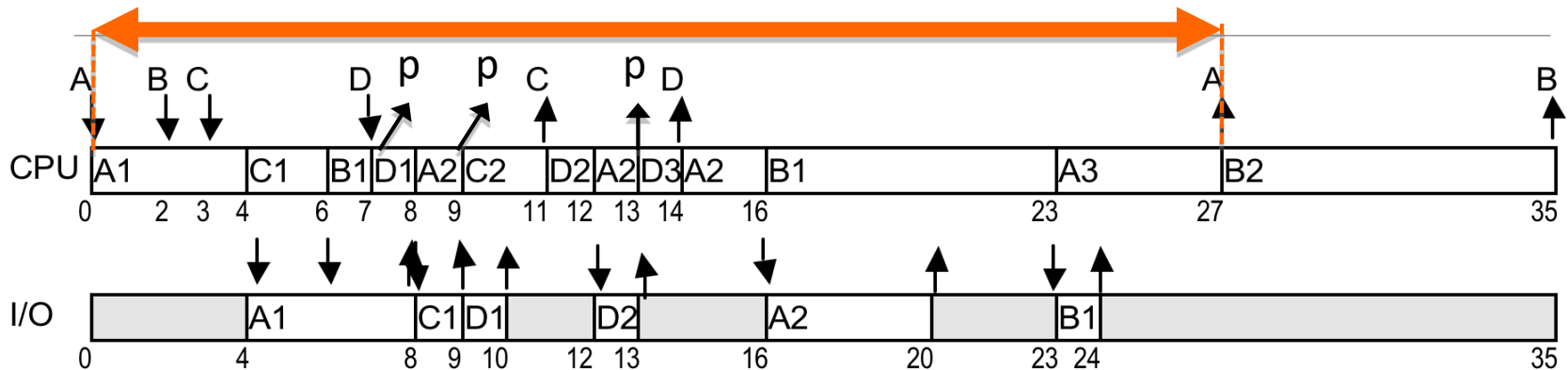


## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Processor utilization =  $(35 / 35) * 100 = 100 \%$
- Throughput =  $4 / 35 = 0.11$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Turn around time:

$$tat_A = 27 - 0 = 27$$

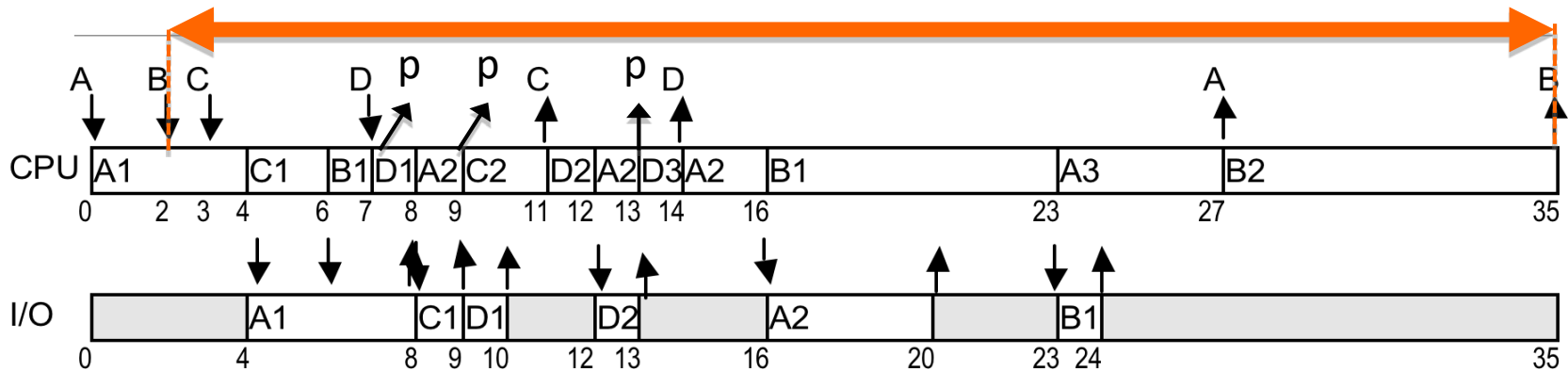
$$tat_B = 35 - 2 = 33$$

$$tat_C = 11 - 3 = 8$$

$$tat_D = 14 - 7 = 7$$

$$tat_{AVG} = (27 + 33 + 8 + 7) / 4 = 18.75$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Turn around time:

$$tat_A = 27 - 0 = 27$$

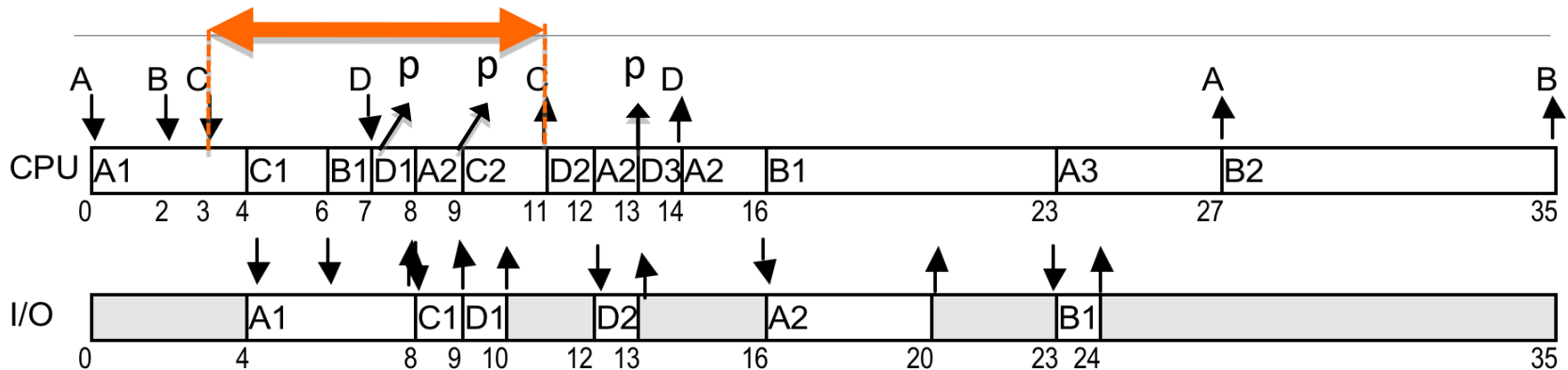
$$tat_B = 35 - 2 = 33$$

$$tat_C = 11 - 3 = 8$$

$$tat_D = 14 - 7 = 7$$

$$tat_{AVG} = (27 + 33 + 8 + 7) / 4 = 18.75$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Turn around time:

$$tat_A = 27 - 0 = 27$$

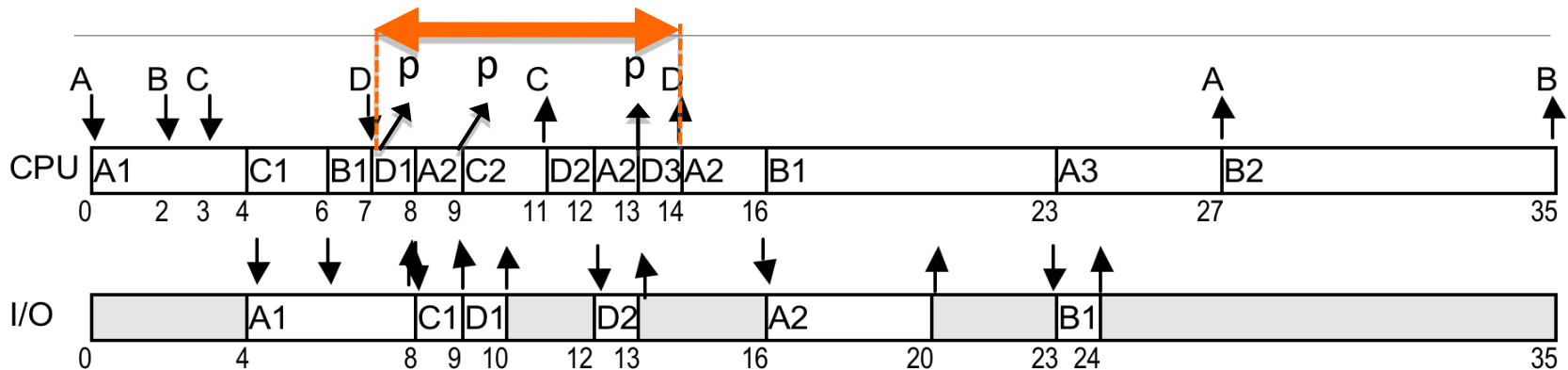
$$tat_B = 35 - 2 = 33$$

$$tat_C = 11 - 3 = 8$$

$$tat_D = 14 - 7 = 7$$

$$tat_{AVG} = (27 + 33 + 8 + 7) / 4 = 18.75$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Turn around time:

$$tat_A = 27 - 0 = 27$$

$$tat_B = 35 - 2 = 33$$

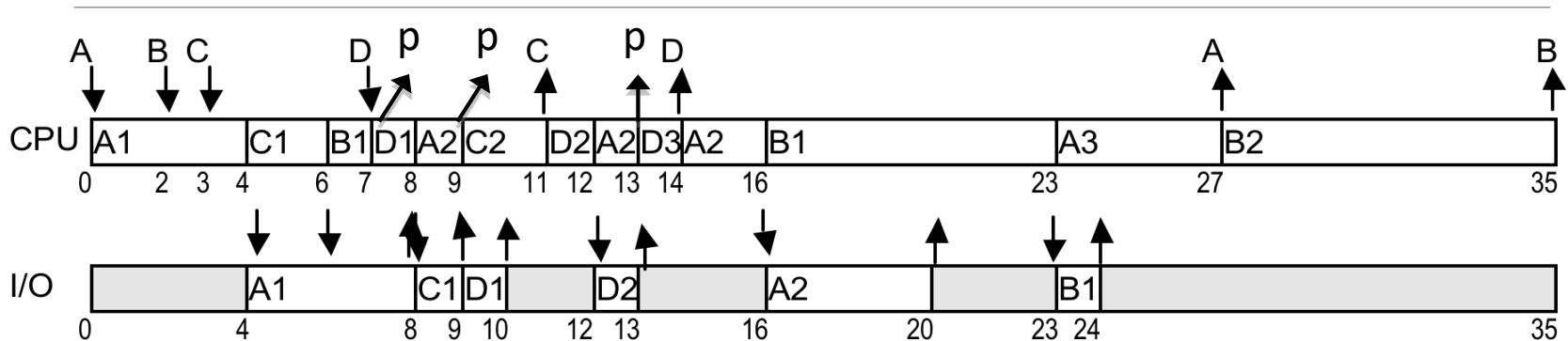
$$tat_C = 11 - 3 = 8$$

$$tat_D = 14 - 7 = 7$$

$$tat_{AVG} = (27 + 33 + 8 + 7) / 4 = 18.75$$



## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Turn around time:

$$tat_A = 27 - 0 = 27$$

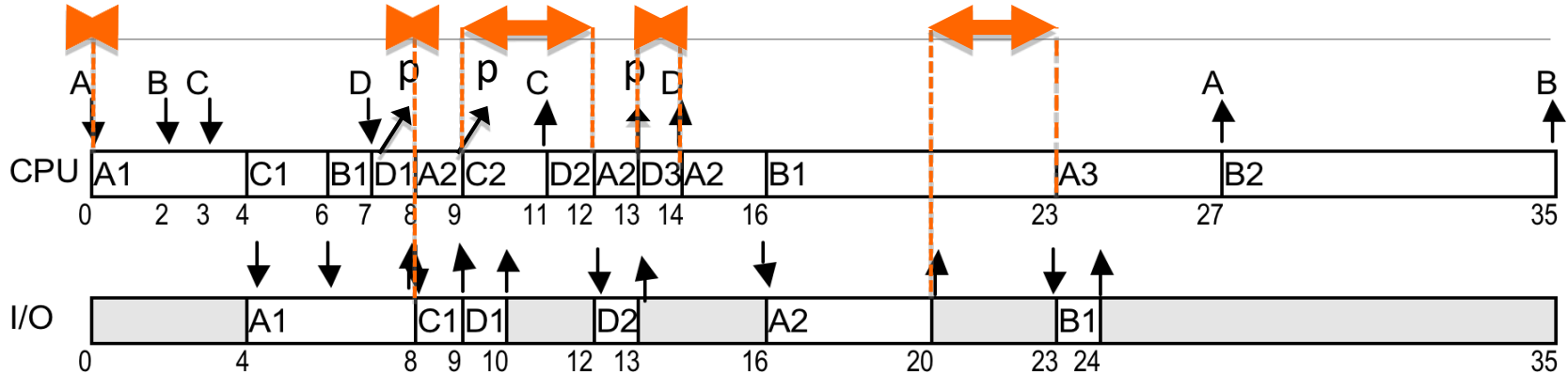
$$tat_B = 35 - 2 = 33$$

$$tat_C = 11 - 3 = 8$$

$$tat_D = 14 - 7 = 7$$

$$tat_{AVG} = (27 + 33 + 8 + 7) / 4 = 18.75$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Waiting time:

$$wt_A = (0 - 0) + (8 - 8) + (12 - 9) + (14 - 13) + (23 - 20) = 7$$

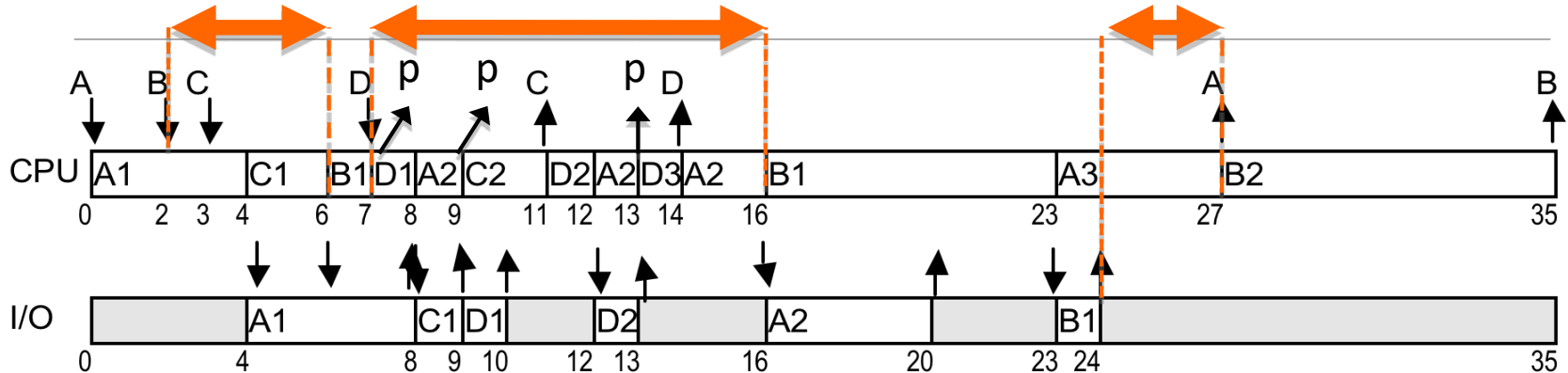
$$wt_B = (6 - 2) + (16 - 7) + (27 - 24) = 16$$

$$wt_C = (4 - 3) + (9 - 9) = 1$$

$$wt_D = (7 - 7) + (11 - 10) + (13 - 13) = 1$$

$$wt_{AVG} = (7 + 16 + 1 + 1) / 4 = 6.25$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Waiting time:

$$wt_A = (0 - 0) + (8 - 8) + (12 - 9) + (14 - 13) + (23 - 20) = 7$$

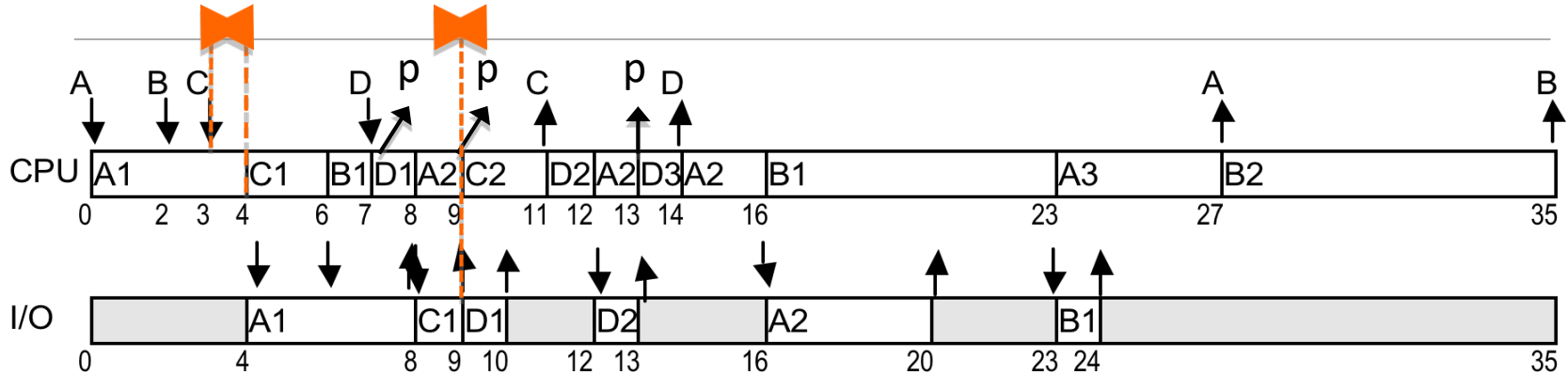
$$wt_B = (6 - 2) + (16 - 7) + (27 - 24) = 16$$

$$wt_C = (4 - 3) + (9 - 9) = 1$$

$$wt_D = (7 - 7) + (11 - 10) + (13 - 13) = 1$$

$$wt_{AVG} = (7 + 16 + 1 + 1) / 4 = 6.25$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Waiting time:

$$wt_A = (0 - 0) + (8 - 8) + (12 - 9) + (14 - 13) + (23 - 20) = 7$$

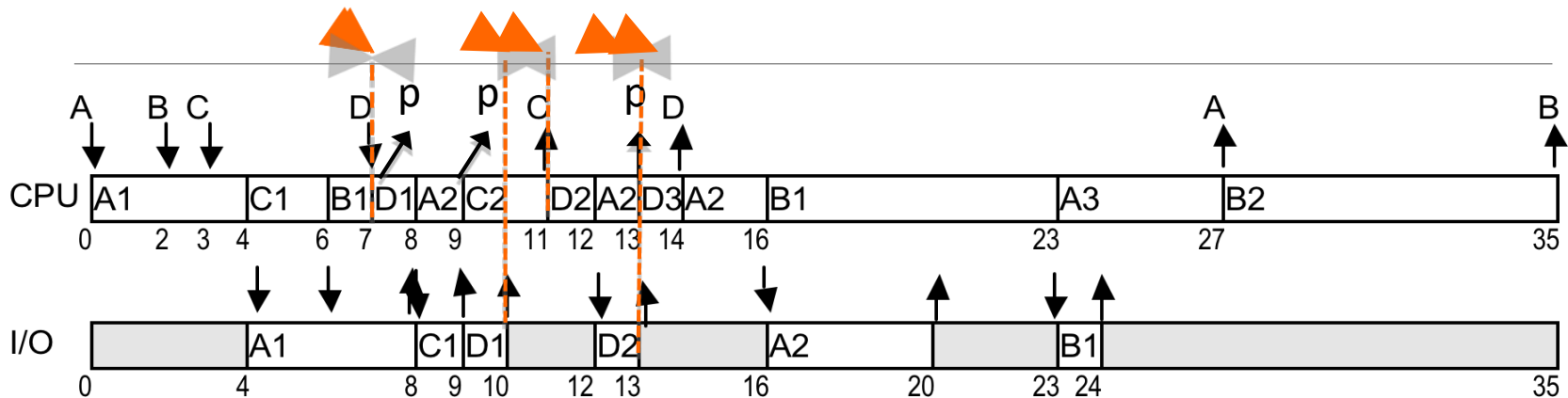
$$wt_B = (6 - 2) + (16 - 7) + (27 - 24) = 16$$

$$wt_C = (4 - 3) + (9 - 9) = 1$$

$$wt_D = (7 - 7) + (11 - 10) + (13 - 13) = 1$$

$$wt_{AVG} = (7 + 16 + 1 + 1) / 4 = 6.25$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Waiting time:

$$wt_A = (0 - 0) + (8 - 8) + (12 - 9) + (14 - 13) + (23 - 20) = 7$$

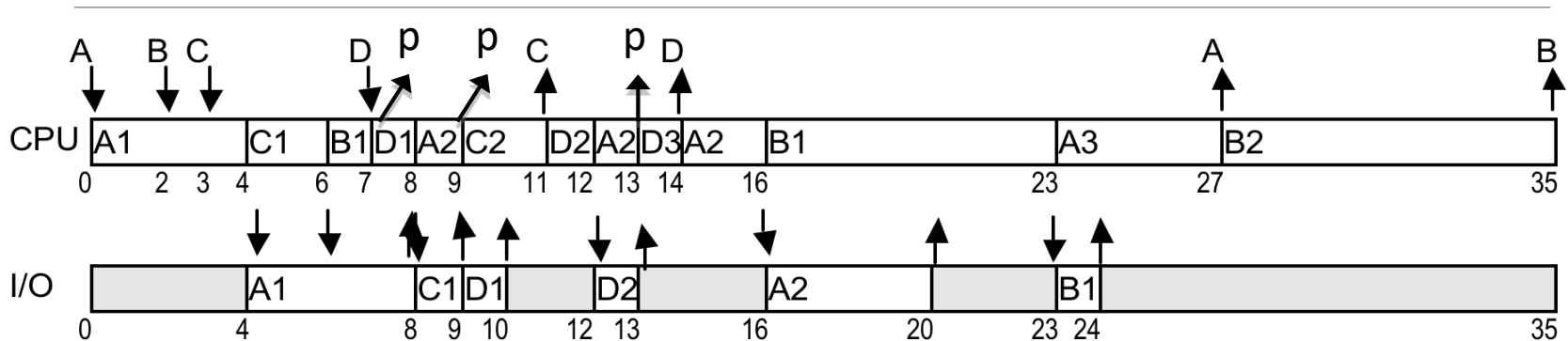
$$wt_B = (6 - 2) + (16 - 7) + (27 - 24) = 16$$

$$wt_C = (4 - 3) + (9 - 9) = 1$$

$$wt_D = (7 - 7) + (11 - 10) + (13 - 13) = 1$$

$$wt_{AVG} = (7 + 16 + 1 + 1) / 4 = 6.25$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Waiting time:

$$wt_A = (0 - 0) + (8 - 8) + (12 - 9) + (14 - 13) + (23 - 20) = 7$$

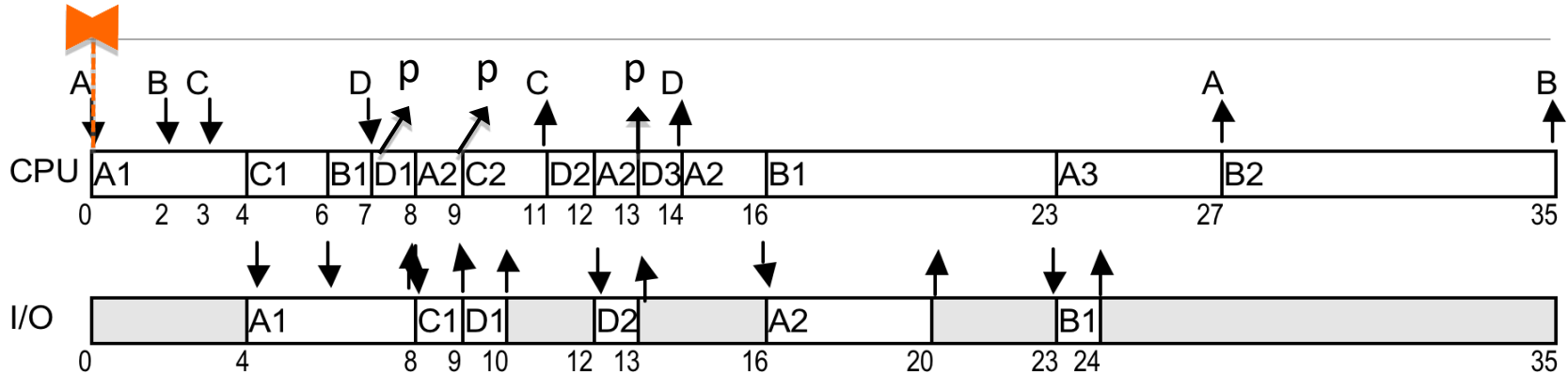
$$wt_B = (6 - 2) + (16 - 7) + (27 - 24) = 16$$

$$wt_C = (4 - 3) + (9 - 9) = 1$$

$$wt_D = (7 - 7) + (11 - 10) + (13 - 13) = 1$$

$$wt_{AVG} = (7 + 16 + 1 + 1) / 4 = 6.25$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Response time:

$$rt_A = 0 - 0 = 0$$

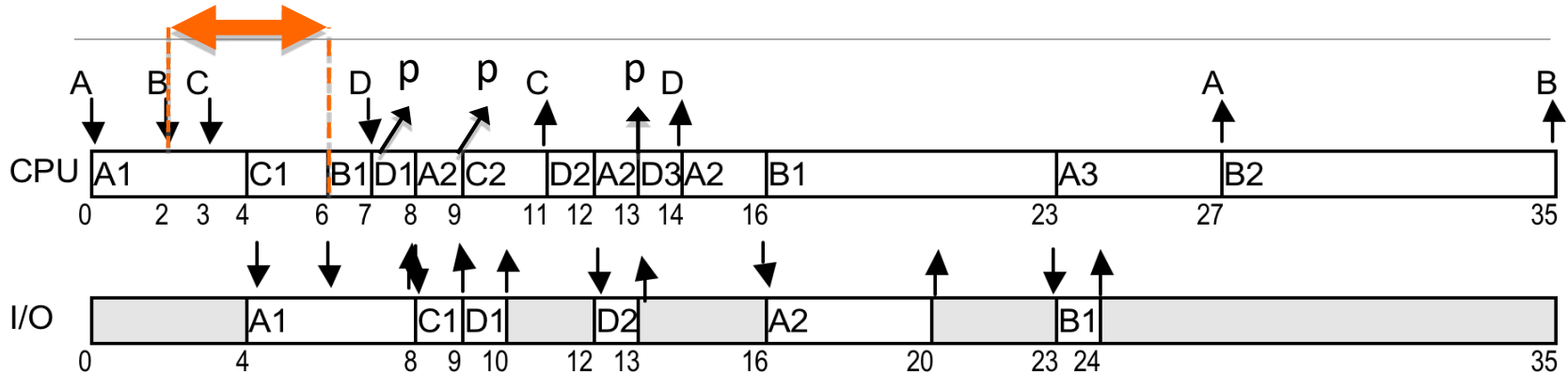
$$rt_B = 6 - 2 = 4$$

$$rt_C = 4 - 3 = 1$$

$$rt_D = 7 - 7 = 0$$

$$rt_{AVG} = (0 + 4 + 1 + 0) / 4 = 1.25$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Response time:

$$rt_A = 0 - 0 = 0$$

$$rt_B = 6 - 2 = 4$$

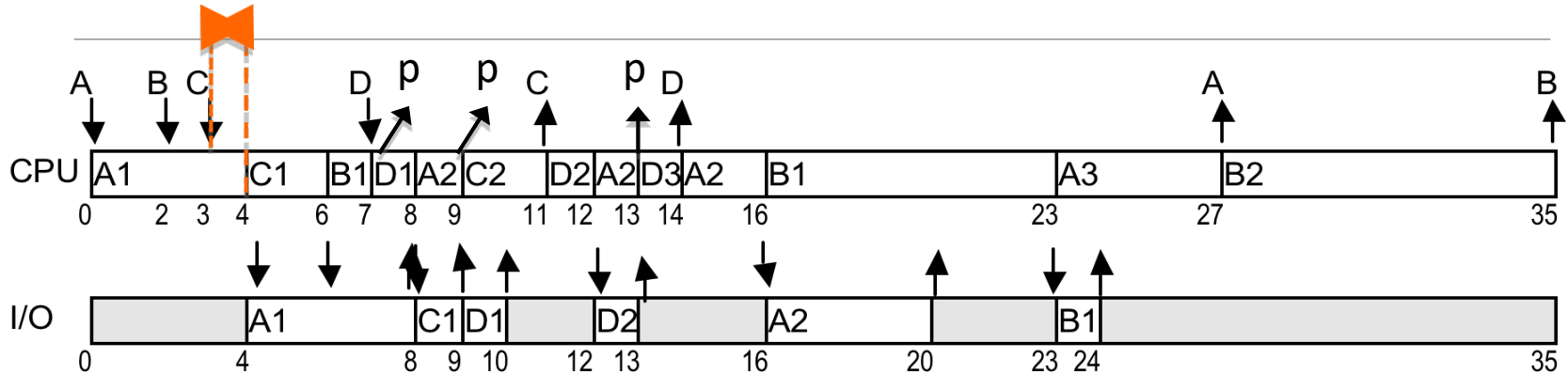
$$rt_C = 4 - 3 = 1$$

$$rt_D = 7 - 7 = 0$$

$$rt_{AVG} = (0 + 4 + 1 + 0) / 4 = 1.25$$



## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Response time:

$$rt_A = 0 - 0 = 0$$

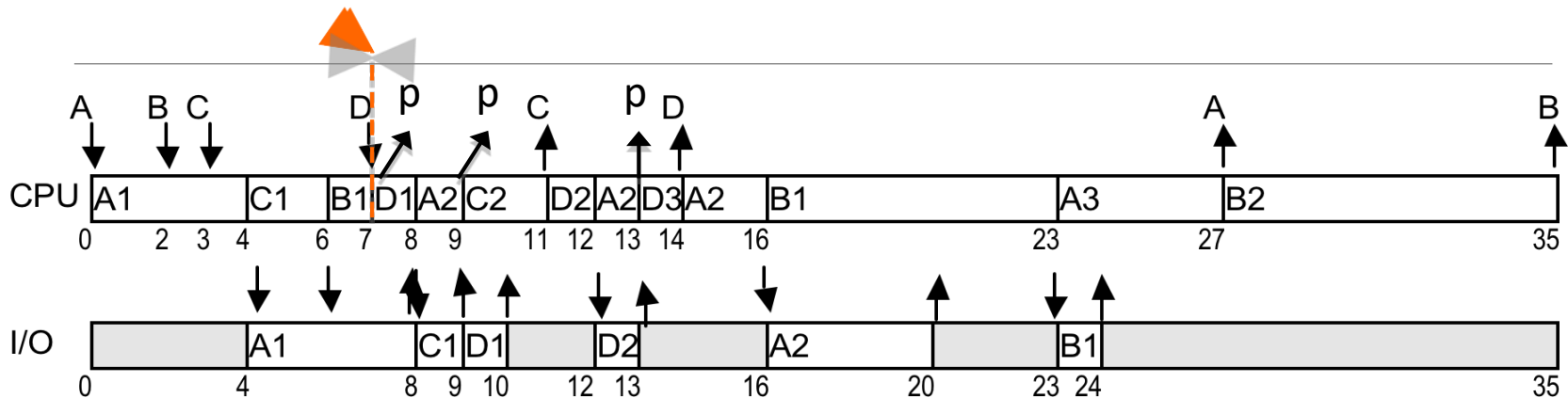
$$rt_B = 6 - 2 = 4$$

$$rt_C = 4 - 3 = 1$$

$$rt_D = 7 - 7 = 0$$

$$rt_{AVG} = (0 + 4 + 1 + 0) / 4 = 1.25$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Response time:

$$rt_A = 0 - 0 = 0$$

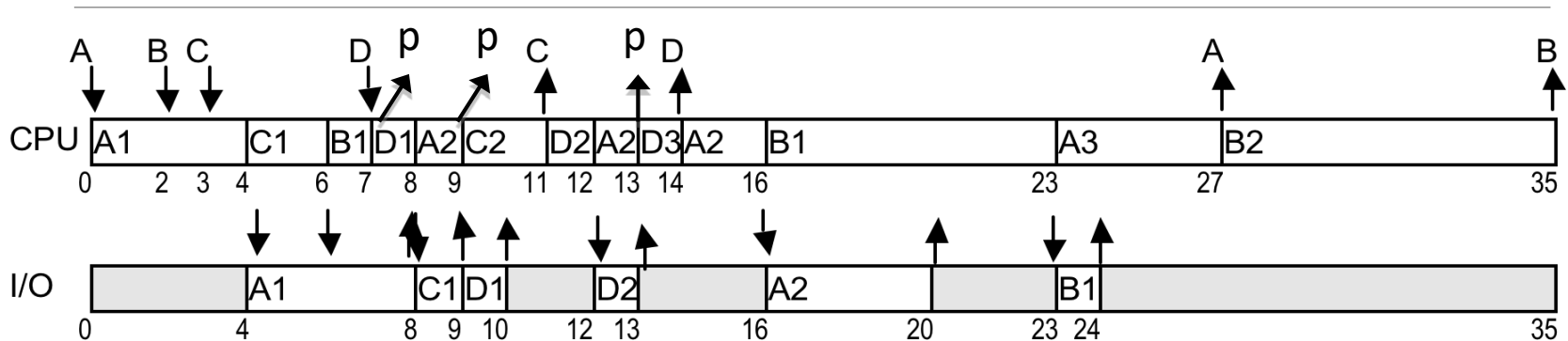
$$rt_B = 6 - 2 = 4$$

$$rt_C = 4 - 3 = 1$$

$$rt_D = 7 - 7 = 0$$

$$rt_{AVG} = (0 + 4 + 1 + 0) / 4 = 1.25$$

## 2.3.3 Shortest-Remaining-Time-First (SRTF)



- Response time:

$$rt_A = 0 - 0 = 0$$

$$rt_B = 6 - 2 = 4$$

$$rt_C = 4 - 3 = 1$$

$$rt_D = 7 - 7 = 0$$

$$rt_{AVG} = (0 + 4 + 1 + 0) / 4 = 1.25$$



# COMPARAISON

	FCFS	SJF	PSJF	Tourniquet
$tat_{avg}$	28.25	23.50	18.75	24.50
$wt_{avg}$	16.50	10.50	6.25	12.25
$rt_{avg}$	4.50	3.00	1.25	2.25
	Facile à implémenter	Impossible de connaître le prochain temps de travail du CPU exacement.	Impossible de connaître le prochain temps de travail du CPU exacement.	Implémentable, $rt_{max}$ est important pour les systèmes interactifs

Tableau récapitulatif